UNIVERSITY OF OSLO
Department of Informatics

Toeplitz Covariance Matrix Estimation for Adaptive Beamforming and Ultrasound Imaging

Master thesis

Michael Pettersson

August 2012

Toeplitz Covariance Matrix Estimation for Adaptive Beamforming and Ultrasound Imaging

Michael Pettersson

August 2012

Acknowledgment

First of all I want to thank my two supervisors, Postdoctoral Fellow Carl-Inge Colombo Nilsen and Associate Professor Andreas Austeng at the digital signal processing and image analysis research group, in the Department of Informatics at the University of Oslo. Carl-Inge has motivated, inspired and guided me towards the art of array processing. He has always been up for discussions, and helped me throughout the thesis. I especially want to thank Andreas, he is one of the reasons why I choose my academic direction of digital signal processing, he has great teaching abilities, and great enthusiasm which has led me into the exciting world of digital signal processing.

I want to thank my fellow students for making the time at the University a joyful period of my life. We will keep in touch and keep the memories alive!

I am also grateful for all the support and patience my family and friends have offered me during this thesis. I owe you all!

Michael Pettersson
Oslo, 31st July 2012

Abstract

In recent years, adaptive beamformers have been researched more extensively, to be able to use it in the application of medical ultrasound imaging. The adaptive beamformers can provide a higher resolution and better contrasts in the resulting images than non-adaptive, and most commonly used, delay-and-sum beamformer. The difficulties of applying adaptive beamformers to ultrasound imaging can e.g. be numerical complexity, stability of statistics, coherent sources, or the robustness of the beamformer. Methods to handle these types of difficulties have been researched and successfully applied to utilize the performance advantages provided by adaptive beamformers.

Why is it important to prevent these types of errors? The beamformer is used in the image formation stage, and errors that occur at this stage are difficult to get rid of even for the most sophisticated imaging software. We have in this thesis investigated methods that attempt to force the estimate of the covariance matrix to become a Toeplitz matrix. A Toeplitz matrix has equal elements along its diagonals, and has several useful properties that are desirable in array processing. Assuming the Toeplitz structure can be achieved, this is because of the spatial stationarity in the received data, it can be applied in medical ultrasound imaging. Three different methods are proposed to reach the desired Toeplitz structure in this thesis; IAA-APES, Adaptive Spatial Averaging and Spatial Convolution. The three methods for making the covariance matrix Toeplitz will be compared and analyzed to other known adaptive beamformers to detect their strengths and weaknesses.

Firstly, this thesis introduces the field of adaptive beamformers. Starting with the basic operations of array processing to build a general understanding of the topic, and continuing with more advanced methods for improving the performance of the array processing.

The second part explains and motivates our Toeplitz constrained methods, and introduces the performance metrics that will be used for comparing and benchmarking the beamformers.

Lastly, we explore the results and discuss the outcome, comparing the Toeplitz constrained methods with other beamformers to benchmark the performance. We apply the Toeplitz constrained methods to the application of medical ultrasound imaging.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

List of Symbols	Definition
$f(\vec{x},t)$	The field at position $\vec{x}$ at time t
$\{f(x_m,t)\}_{m=0}^{M-1}$	The field at every m'th sensor
c	Speed of light/ propagation
x_m	Sensor location m
$\vec{y}\,[n]$	Output data
$\hat{\vec{y}}$	Zero-padded output data
$\vec{z}\,[n]$	Weighted output
$\vec{w}$	Filter coefficients (weights)
M	Number of sensors
N	Number of time samples
L	Number of elements in a subarray
K	Number of subarrays
k	Index of subarray
G	Number of scan lines (angle resolution)
$\mathbf{R}$	The true correlation matrix
$\hat{\mathbf{R}}$	The estimated correlation matrix
$\hat{\mathbf{R}}_{sa}$	Correlation matrix based on subarray averaging
$\hat{\mathbf{R}}_{sc}$	Correlation matrix based on spatial convolution method
$\hat{\mathbf{R}}_{asa}$	Correlation matrix based on adaptive spatial averaging
APES	Amplitude and Phase EStimation
DAS	Delay and Sum
DOA	Direction of Arrival
IAA-APES	Iterative adaptive approach for amplitude and phase estimation
FIR	Finite Impulse Response
MPDR	Minimum Power Distortionless Response
MVDR	Minimum Variance Distortionless Response
ULA	Uniform Linear Array
SNR	Signal to Noise Ratio
RMSE	Root Mean Square Error

Part I

Introduction

Chapter 1

Background

1.1 Technology Development

The technology development has been driven by the curiosity of exploring beyond mankind's limitations. The great astronomer Galileo Galilei is an example of eminent exploration beyond what had been done. He made improvements on the telescope and explored the moons around Jupiter, and he was not stopped by the limitations of the human eye. To be able to exceed mankind's limitations, scientists have studied and copied nature to gain greater understanding. This is also the case for *digital signal processing*; natural sensors have been studied to enhance the performance of artificial sensor systems. Humanity is equipped with natural sensors such as eyes and ears. With our eyes we are able to detect electromagnetic radiation in the approximate frequency region 400- 750 THz, which is the color spectra or also called visible light. For humans, hearing is normally limited in frequencies between 20- 20 000 Hz, however these limits are not definite, and the upper limit declines with age. We are all band limited with our natural sensors, which has caused creativity to explore beyond these limitations. We are now able to "see" with sound, by using sound waves to create images because of curiosity of scientists. With sound waves we can penetrate including the human body to create images from the inside of the body without the risks of medical surgery. To be able to use sound waves as eyes, we copy the processing chain of the natural sensors. Our natural sensors are able to detect and filter signals, and the signals are sent to our processing unit, the brain, for interpretation. The brain is able to respond fast and make the right decision. These unique processes have saved many lives from dangerous situations throughout the world's history. We can proudly mention that digital signal processing has already saved lives, within medical, communication, and defense applications. Digital signal processing is based on the same techniques as our natural sensors, throughout this thesis we will do sampling, filtering, and interpretation of signals. To make this possible many of the greatest scientists have contributed to the art of digital signal processing with mathematical tools and understanding of the fundamental physical processes. Heinrich Hertz is one of them; he was the first that demonstrated the existence of

electromagnetic waves, by building an aperture that could produce and detect radio waves. This demonstration can be seen as a new generation of applications that can go beyond the limitations of the natural sensors. As a result of digital signal processing we have wireless communication, remote sensing, medical imaging, and other digital signal applications. These exploit wave technology, with advantages such as; speed of propagation, penetration and distance, as information carriers.

1.2 Wave Propagation

Electromagnetic (EM) and pressure (acoustic and seismic) waves are information carriers of the modern world. We are surrounded by these waves every day, due to the modern world technologies. It was the Scottish physicist and mathematician James Clerk Maxwell that described the EM waves mathematically and speeded up the technology development. He derived the Maxwell equations, which relate the electric field and the magnetic field, and how they together form a propagating wave. The wave equation can be derived from Maxwell's equation, and the resulting equation is a second order linear partial differential equation;

$$\nabla^2 s = \frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial y^2} = \frac{1}{c}\frac{\partial^2 s}{\partial t^2} \tag{1.1}$$

With this partial differential equation we are able to describe the propagation of all types of waves in different spatial environments, whether it is electromagnetic or pressure waves. To use waves as information carriers we need to have a-priori knowledge about the environmental conditions. The environmental conditions will affect the wave's propagation, whether it is physical obstacles or changes in the speed of propagation, c. The success criterion relies on how accurate the characteristics of the environment are being measured. If the environment model is made close to perfect, we are able to extract the desired information from the traveling waves without difficulties.

1.3 Signals

A simple narrowband signal can be described as;

$$s(t) = A\cos(2\pi f_c t) \tag{1.2}$$

where f_c is the *center frequency*, t defines the *time* and A is the amplitude of the signal. The signal is often represented in a complex and compact form called the Euler formulation;

$$s(t) = Ae^{j2\pi f_c t} = A\cos(2\pi f_c t) + Aj\sin(2\pi f_c t) \tag{1.3}$$

To generate an "A" tone, we set the frequency to $f_c = 440Hz$ with a given time duration t as arguments. The signal will create disturbance

in the environment in terms of pressure waves that propagates outward from the signal source. The same effect can be illustrated as a stone tossed in calm water where the surface waves will propagate outward from the source of disturbance, and make multiple circles in the water. Now we have illustrated a signal that propagates in space, which will introduce the third dimension *space*, in addition to what we already have, *time* and *frequency*. To describe a propagating signal in space domain we include *spatial coordinate* $\vec{x} = (x, y, z)$, and *spatial frequency/wavenumber* $\vec{k} = (k_x, k_y, k_z)$;

$$s(\vec{x}, t) = Ae^{j\left(2\pi f_c t - \vec{k}\vec{x}\right)} \tag{1.4}$$

$\vec{k}$'s magnitude represents the number of waves per meter $k = |\vec{k}| = \frac{2\pi}{\lambda}$, and its normalized vector give us the direction of propagation. A signal propagating through space with a frequency f_c at a speed of c will give us the *wavelenght* $\lambda = \frac{c}{f_c}$. Thus, $\vec{x}$ and $\vec{k}$ are dual variables in the same sense that t and $2\pi f_c$ are in one dimension.

1.3.1 Wave Front

Disturbance generated by a source make periodes radiate circularly outward from the source, whether it is surface or spatial waves depending on the substance the wave is propagating in. As the wave propagates with a distance r, the wavefront size will increase. When sampling the wave, the interesting feature of the wave is the shape of the wavefront, because the calculation can be simplified if the wavefront is known. The wavefronts are classified into two different categories, curved and plane wavefront. The two different wavefronts are related to the traveling distance r, and the size of sampling area D. It is often separated into near and far field, or the approximations called Fresnel and Fraunhofer.

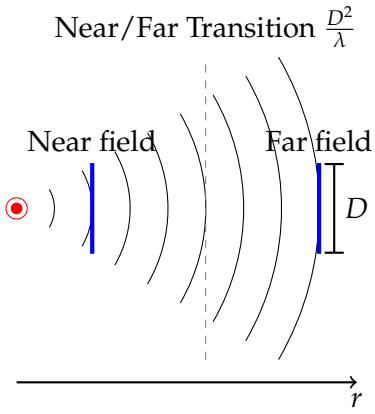


Figure 1.1: Near/Far-field transition

The simplest form of wavefront is a wave that can be interpreted as a plane wave, which occurs in the far field. For a plane wave the phase difference is linear, and easy to calculate because it only moves only along one axis. The term plane wave arises when the spatial plane $\vec{x}$ has the same

value at any arbitrary time t. The threshold where a spherical wave can be interpreted as a plane wave is called the near/far field transition $\frac{D^2}{\lambda}$. D is the size of samplings area or aperture, and $\lambda = \frac{c}{f}$ is the wavelength, which describes the spatial period a wave use to repeat itself. A plane wave is be formulated as

$$s(\vec{x}, t) = Ae^{j\left(2\pi ft - \vec{k}\vec{x}\right)} \tag{1.5}$$

The plane wave is simpler than spherical wave front, but there are several important benefits of working in the near field instead with spherically wave fronts. In the near field is it possible to detect both the distance and direction of a source. In applications such as medical imaging, location is essential to get the correct position of different areas of interests, such as a cyst. When formulating the spherical wave equation, it is convenient to change to spherical coordinates, given by r, ϕ, θ. For spherically symmetric problems, the spherical wave equation is not depended on the angles ϕ and θ, and becomes

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial s}{\partial r}\right) = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2} \tag{1.6}$$

One solution to the general spherical wave equation is the monochromatic one:

$$s(r, t) = \frac{A}{r}e^{j\left(2\pi ft - \vec{k}r\right)} \tag{1.7}$$

The spherical wave is propagating outwards from the origin,and r is the distance the wave has traveled.

1.3.2 Wave Fields

This thesis concentrate on signals in the field received by several sensors. The energy in the wave field is converted by sensors into digital form, such that signals that propagates in the environment can be recorded. To record the observations from a specific location in the wave field $f(\vec{x}, t)$, we simply put a sensor at same location of interest. By placing M sensors out in the field $\{f(x_m, t)\}_{m=0}^{M-1}$ we cover a larger area of the field. Where x_m describing the position of the sensor m, and t is the time when the field is passing the sensor. We aim to sample the field, and extract wanted information gathered by the sensors. Each sensor makes an output sample $y(x_m, t)$ of the field for the given sensor position x_m and every t_n. The sensor outputs made out of the field consists of an *additive combination of signals* and *noise*. When the wave field is converted from analog to digital representation at each sensor, some disturbance is being applied to the measured field. Disturbance such as thermal noise or noise from surrounded electronics are found in the digital representation. This noise is a stochastic sequence, usually stationary, zero-mean Gaussian, and is statistically independent of the field's signals. Uncorrelated noise is easy to remove from the signal, and in some applications desired. The desirable form of noise is a signal with zero-mean and with flat power spectrum is called *white noise* (1.2), because it is stationary and uncorrelated with the signals. The noise is

randomly generated and has no information, and can interrupt the desired signal if the noise level is high.

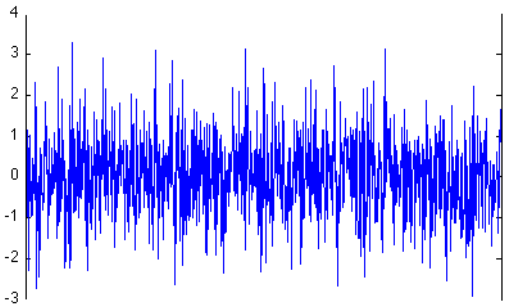


Figure 1.2: Randomly generated noise

The stationary assumption means that the noise has the same statically characteristics on every sensor output, and are constant over time. Any cross-correlation between the sensor outputs relies on the distance between the sensors, which might make the signals difficult to separate. Noise can also arise when we have interference of other signals that are not of interest.

These signals are called *jammers* or *interfering* signals. Interference is a phenomenon that occurs when two or more signals overlap in the same region of space. Interfering signals that propagate toward our sensors can disturb our algorithms, by focusing the beam in the undesirable direction. In many implementations of array processing the signals are assumed to be narrowband. Resulting in a baseband sensor output at location m'th, and can be expressed as a linear combination of *desired signal stationary noise and interfering signals*.

$$y(x_m, t_n) = \sum_{n=0}^{N-1} \overbrace{s(x_m, t_n)}^{\text{wanted}} + \overbrace{(i(x_m, t_n) + p_m(t_n))}^{\text{unwanted noise}} \tag{1.8}$$

This equation gives the output on the m'th sensor before any kind of preprocessing. With our sensors spread correctly[1] out in the field. When combining outputs from several sensors we are able to do filtering in spatial domain. The sum of the total array of M elements is defined as;

$$z[n] = \frac{1}{M} \sum_{m=0}^{M} \vec{y}(x_m, t_n) \quad n = 1, \cdots, N \tag{1.9}$$

Given the opportunity of filtering in the spatial domain, it is possible to find the direction of a source. The direction can be calculated with the time delay or the phase different between the sensors perceiving the incoming wave from the radiating source.

A useful statistics tool for measuring the field is the correlation function on the sensors outputs. The resulting correlation matrix play a key role

[1] Nyquist-shannon sampling theorem

when implementing the beamformers. Correlation is a statistical concept that describes the degree of relationship between two random variables. In array processing it is used the check the relation between the sensors, they should sense the same but not at same time or place, assumed that the array is statically positioned. We denote the true correlation matrix as $\mathbf{R}$, and denote our estimation of $\mathbf{R}$ as $\hat{\mathbf{R}} = E\{\vec{y}[n]\,\vec{y}[n]^H\}$ which will appear throughout this thesis. The field we sample almost always contains artifacts that affect the estimate of the correlation matrix. The artifact is coherent signals. Coherence is an effect that causes correlation between the desired signal and one or more inferring signals. If interference and desired signals are uncorrelated, the beamformer are able to attenuate the interference to a minimum. Otherwise, when sources are coherent it may lead to signal cancellation or a portion cancellation of the signal of interest. The coherence appears when plane waves have common phase reference. This phenomena is well known for the minimum variance beamformer, and other optimization beamformers. Looking at signal statistics of an array, we may analyze the spatial covariance to make simplified assumptions of the wave field. Coherence can arise due to multi-path propagation, when the desired signal and the inference signal are equal but arrive from different angle. Coherent interference signal has the same source as the desired signal, but has arrived through a different path. When waves are propagating through a room the walls will reflect the waves, and the original and reflected wave will then have the same signal content but arrive from different angles when the sensor measures it. This is an example of coherent interference. The cross

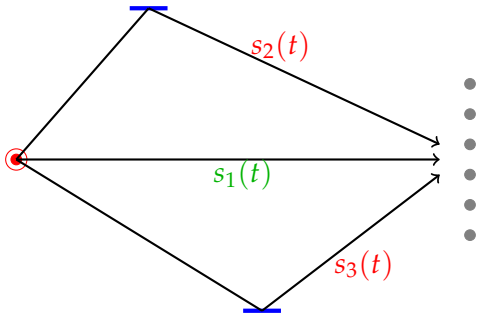


Figure 1.3: Illustration of multi-path propagation

correlation between coherent interference created by multi-path and the desired signal is nonzero since the phase relation stays constant. Because of this, the directionally constrained beamformer minimize the power and will interpret these two waves as one. If the interfered signal is fully correlated to the desired signal it will result in signal cancellation. When we only have partial correlation the beamformer will only cancel portions of the desired signal that correlates with the interfered signal. To get rid of the artifact of correlated sources we can decorrelate our estimates with subarray averaging or filtering of the covariance matrix. These methods will be described in detail later in the thesis.

1.4 System Setup

In this section we will briefly describe each component and the procedure to be able to form beams. This include the basic operation describing the beamformer system mathematically and the different assumption we have to consider.

1.4.1 Sensor

We have two different types of sensors, directional and omnidirectional. A parabolic radar dish is an example of a directional sensor, because it only receives signals from one direction. The geometrical design of a parabolic dish is beneficial due to pre-focusing of the incoming waves for the sensor, to change the direction/focus we have to physically move the parabolic dish. In contrast to geometrical pre-focused sensors, we have omni-directional sensors that are emitting the wave field power uniformly in every direction. With an omnidirectional sensor it is not possible to decide the direction of the incoming wave.

An aperture is used to sample the wave field, the aperture is described by its spatial shape and extent. The aperture shall convert energy from the field $f(\vec{x_m}, t)$ at the sensor location $\vec{x}_m$ into energy that can be interpreted on a computer, like a transducer. To gather the particular field energy the aperture spatially integrates the energy over the apertures $w(\vec{x})$ spatial extent. The aperture is the sensor, and to describe our sensors mathematically we use the aperture smoothing function;

$$W(\vec{k}) = \int w(\vec{x}) e^{j\vec{k}\vec{x}} d\vec{x} \tag{1.10}$$

The surface of the sensor is continuous and described by the aperture function, with integration we can calculate the aperture surface. A sensor can have geometric extent in all three dimensions, (x, y, z). A perfect aperture smoothing function is the three-dimensional impulse function $W(\vec{k}) = \delta(\vec{k})$, this is not feasible to obtain, because we can not achieve a sensor with unlimited extent in space. The task of the sensor is to emit signals, the sensor itself can not determine the range or direction. In general an aperture $w(\vec{x})$ with a large spatial extent has a narrow aperture smoothing function $W(\vec{k})$. The spatial extent of an aperture determines the angle between two sources this is normally called *resolution*. The larger the aperture the more focused the aperture can be in any specific direction.

A common measurement of an aperture performance is directivity. Directivity is the ratio between the estimated power in the desired direction $P(\theta_d, \phi_d)$ divided by the power estimate for all possible directions of arrival. In general term directivity is given by;

$$DIR_{2D} = \frac{P(\theta_d, \phi_d)}{\frac{1}{4\pi} \int_0^{\pi} d\theta \int_0^{2\pi} d\phi \sin(\theta) \cdot P(\theta, \phi)} \tag{1.11}$$

In our case we shall focus on sensor elements placed on one axis, then ϕ

dimension is disregarded.

$$DIR_{1D} = \frac{P(\theta_d)}{\frac{1}{2}\int_0^{\pi} P(\theta)sin(\theta)d\theta} \tag{1.12}$$

1.4.2 Array

The word array has its origin from the old French word *areer*, which means *" to put in order"*. An array consists of a group of sensors, directional or not, combined to form a single output. The array samples the environment spatially, so the sampling theorem [2] has to be followed to avoid aliasing. There is not much difference between the time-frequency response and the spatial frequency response. In time-frequency signal processing the filter commonly used is stated as;

$$H(e^{j\omega T}) = \sum_{m=0}^{M-1} h_m e^{-j\omega Tm} \tag{1.13}$$

The filter above is a linear shift-invariant filter, weighted with impulse response h_m and sampled with spacing equal to T. The array pattern is formulated

$$W(\vec{k}) = \sum_{m=0}^{M-1} w_m e^{j\vec{k}\cdot\vec{x_m}} \tag{1.14}$$

The array pattern describes $m = 0, 1, ..M-1$ sensor outputs, and stacks them together to form one output. The w_m are the complex weights of each sensor, each sensor can either be weighted equally or have individually weighting. $\vec{k}$ is the wavenumber that describes the steering direction. $\vec{x_m}$ is the position of the m'th sensor. If we have ten equally spaced sensors in a linear grid, the array pattern can be calculated. Using more than one sensor has the advantage that the noise can easily be removed, because the noise is unique at each sensor. When we arrange sensors in a geometrical structure to form an array, we can then steer to the focus direction due to the beamforming. The direction of arrival of a single point source is found by directing a spatial filter towards the signal of interest. When we have calculated aperture smoothing function for one sensor we can calculate the array pattern to place sensors in the spatial domain. Calculation of the array pattern can be done separately using point sources instead of the aperture function for the sensors. After calculating the aperture function and the array pattern, we combine them to obtain the array response;

$$W_{total} = W_{sensor} \cdot W_{array} \tag{1.15}$$

When we use several sensors we have more control over the noise, and are then able to accept some errors/noise since other sensors can smooth out the error so that the noise is reduced. When placing sensors in the spatial domain, all sensor position $\{x_m\}_{m=0}^{M-1}$ are relative to the phase center. For a *uniform linear array (ULA)* is it normal to have the phase center in

[2] Nyquist-shannon sampling theorem

the middle of the array. The phase center is the "center" of the array, not necessary be at the origin. The delay between signal registrations is calculated relative to the phase center of the array, and this can be used to calculate the angle of arrival. The sensors in the array can be arranged in many different geometrical structures depending on the system needs. The array structure can be varied and more complex, but when working in the far-field with plane waves, its is relative simple to calculate the steering weights compared to near-field. Curved wavefront will increase the complexity and requires more calculation. To use this advantage of having more than one sensor, we have to align the gathered signals with the sensors so they can be summed. This method is called *delay and sum*.

1.4.3 Uniform Linear Array

A uniform linear array (ULA) is the simplest array due to its geometrical structure. The array elements are placed along one axis with uniform spacing d. The center of the array is placed at the origin of the coordinate system, for computational and notational convenience. This center is also denoted as the phase center and will be used throughout the text. To sample the field correctly with our ULA we have to fulfill Shannon-Nyquist sampling theorem. The minimum spacing between the sensor elements are determined to avoid under-sampling[3] of the signal with a given frequency is defined as;

$$f = \begin{cases} \lambda/2 & d \leq \lambda \\ d/2 & d > \lambda \end{cases} \tag{1.16}$$

where λ is the wavelength of the signals of interest. With array elements placed along one axis it is possible to detect sources in the θ angle. The visible region to our array will be $-\frac{2\pi}{\lambda} \leq k_x \leq \frac{2\pi}{\lambda}$. When sampling the field uniformly with an ULA it can be compared to regular sampling in time-domain, and apply the same techniques to process the data. When increasing the dimensional extent of the array is it possible to detect sources in θ, ϕ and r. We divide array geometries into three categories linear,planer,and volumetric (3D) arrays. Within each category the field can be sampled; uniformly,non-uniformly, and random according to the sensor positions. In our case we concentrate on a uniform linear array.

1.5 Beamforming and Spatial Filter

Beamforming is the name given to array processing algorithms, where the task is to steer and focus the beam to capture or transmit energy from one specific location. A beam is formed toward a user-selected "look-direction" by the beamformer. The concept of beamforming is to accentuating or attenuating signals from specific directions to help distinguish between signals of interest and interfering signals from unwanted directions. It is desired to gain the ratio between the desired direction and undesired directions. An array of sensors sampled the wave field as a function of time

[3] aliasing

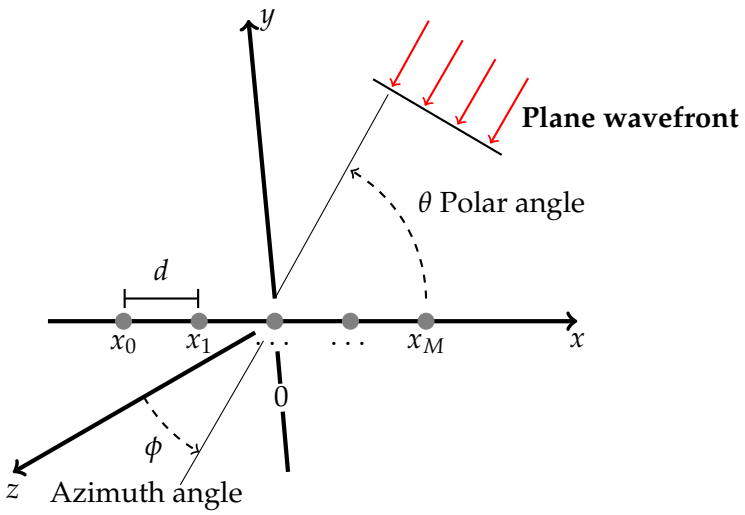
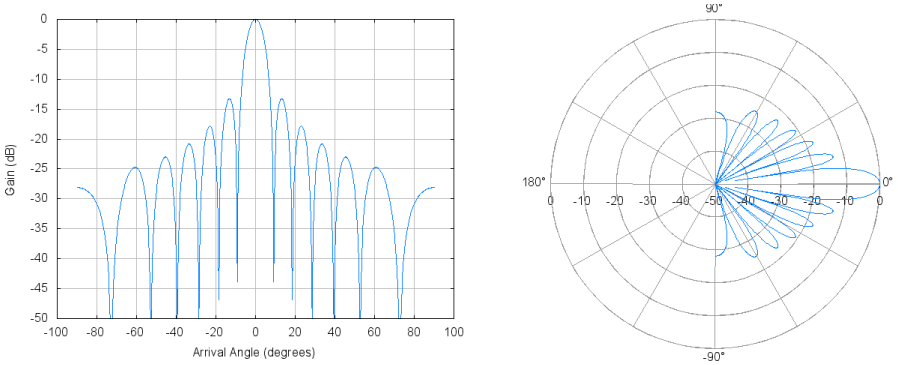


Figure 1.4: Uniform linear array setup, with a plane wave arriving.



(a) Steered response

(b) Polar steered response

Figure 1.5: Steered response, with a source in 0°.

and spatial domain, is necessary because of the discrete spatial extent of the array, makes it able to do filtering in the spatial domain. Spatial filtering can separate signals with overlapping frequency that originate the array from different angles. To make a temporal filter it requires processing of data collected over a temporal aperture. Similarly, for making a spatial filter it requires processing of data collected over a spatial aperture. After filtering, the mainlobe is pointing in the direction of interest, and the thickness of the mainlobe decides the resolution. The sidelobes are energy that are suppressed and can be refereed as noise level.

1.5.1 The Estimated Covariance Matrix

The covariance matrix describes the statistical relationship between two random variables in a vector, and indicates how the two variables change together

$$\mathrm{Cov}(X, Y) = \mathrm{E}\{(X - \mathrm{E}\{X\})(Y - \mathrm{E}\{Y\}\} \tag{1.17}$$

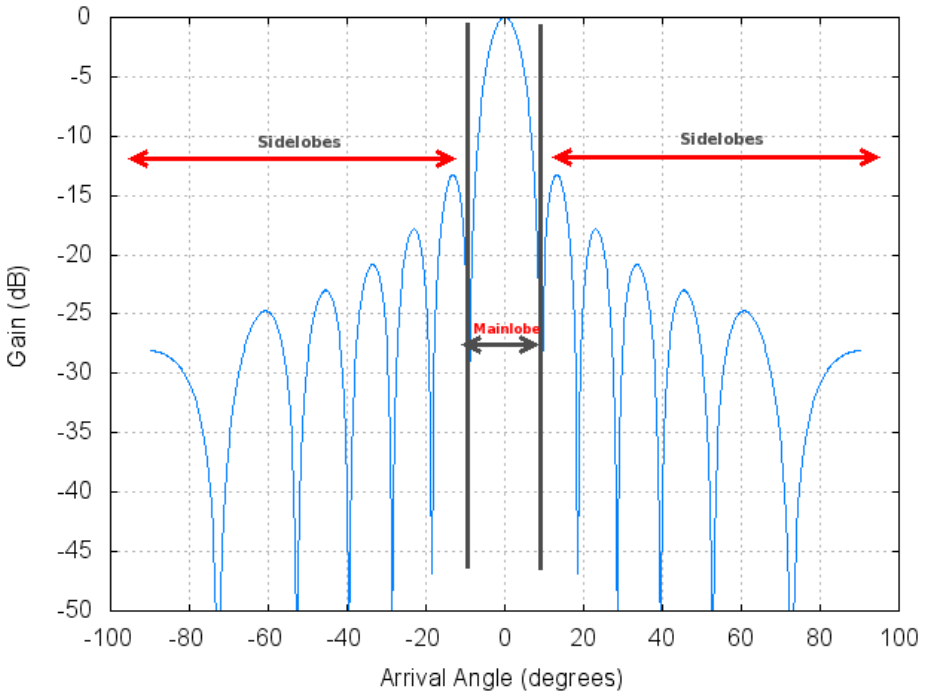


Figure 1.6: Beampattern expalnation

Where $\mathrm{E}\{X\}$ is the expected value of X. By simplifying the equation (1.17) by using of the linearity property of expectations, we have then;

$$\mathrm{Cov}(X, Y) = \mathrm{E}\{XY\} - \mathrm{E}\{X\}\mathrm{E}\{Y\} \tag{1.18}$$

The sampled covariance matrix is commonly used in array processing algorithms to get the statistical relation between every sensor output. It is derived from the power estimate of the sensor outputs under the assumption of zero-mean.

$$P = \frac{1}{N}\sum_{n=0}^{N} |\vec{y}_m[n]|^2 = \frac{1}{N}\sum_{n=0}^{N} \vec{y}_m[n]\,\vec{y}_m[n]^H = \mathrm{E}\{|\vec{y}_m[n]|^2\} = \hat{\mathbf{R}} \tag{1.19}$$

The notation $\{\cdot\}^H$ defines complex conjugate and transpose, the combination of the two operations are often referred as Hermitian. The estimation of the covariance matrix $\hat{\mathbf{R}}$ depend on the data quality, and it is essential for the accuracy. The statistics is not reliable with few data samples. When $\hat{\mathbf{R}}$ is defined we have the opportunity to manipulate the array outputs by filtering, to improve the performance of the beamformer.

1.5.2 Delay and Sum

The method *delay and sum (DAS)* is done by delaying the sensor output, the signals is then aligned so that they can be summed. This is done in relation to the direction we want to steer. The delay Δ_m aligns the incoming signals for the m'th sensor. For a monochromatic wave the delay is a

linear function. Delay and sum is the simplest beamformer and also the most commonly used, sometimes under the names "back projection" or "conventional beamformer". It is most used because of the computationally simplicity and very robust. The algorithm sums up all of the sensors, for every possible angle of arrival. If we correctly sample a narrowband signal at two different locations in space the phase will change during its propagation. This is the concept behind phased array antennas. Instead of having a system mechanically sweeping the directions through space, the antenna can vary the phase electronically and produce a moving pattern without any moving parts. Adjusting the delay with the phase instead of the time gives us high angle resolution even though we only have a few time samples. The weights for the delay and sum beamformer are also predetermined so any further calculation is not needed. The weights for delay and sum beamformer can be used to increase the ratio between signal and noise at the expense of resolution. Many other beamformers are extensions of delay and sum, which is sometimes called the conventional beamformer. Applications of the simplest bearmformer, delay and sum, include radar,sonar,seismology,wireless communications and speakers. As more powerful computers are developed, other methods are going to replace the DAS beamformer. Delay and sum beamformer can be described mathematically as sum over all sensor inputs.

$$\vec{z}\,[n] = \sum_{m}^{M} w_m^H \vec{y}_m\,[t_n - \Delta_m] \tag{1.20}$$

where y_m are the sensor output and each sensor has it own sensor delay Δ_m. The weights w_m^H consist of complex and transpose (hermitian) weights, with these weights we can manipulate the sensor outputs. When we choose to steer our array of sensors, we change the phase according to the delay for each sensor. This is done to stack all signals together, to form one output. To find the power spectrum from the beamformer output we use the following equation $P = \frac{1}{N}\sum_{n=1}^{N} |\vec{z}\,[n]\,|^2 = E\{|\vec{z}\,[n]\,|^2\}$

$$\begin{aligned} E\{|\vec{z}[n]|^2\} &= E\{|w_m^H\vec{y}[n]|^2\} = E\{w_m^H\vec{y}[n](w_m^H\vec{y}[n])^H\} \\ &= E\{w_m^H\vec{y}[n]\vec{y}^H[n]w_m\} = w_m^H E\{\vec{y}[n]\vec{y}^H[n]\}w_m = w_m^H\hat{\mathbf{R}}w_m \end{aligned}$$

$\hat{\mathbf{R}}$ was introduced in the equation above, referring to the estimated correlation matrix. Delay and sum uses predetermined weights, may be applied to reduce the numerical complexity. When filtering the amplitudes with a predefined filter w_m, this permits trading of mainlobe width against the sidelobes levels to form the response into a desired shape. Beamforming is all about forming beams, by gaining the desired direction of arrival and suppress the rest. This classical beamforming are the weights independent from the received data, and often we assume that signal is narrowband, and can be compared to typical Finite Impulse Response (FIR) filtering.

Many different standard filters exits, and in general filters are chosen with the criteria of;

- How wide and how much ripples is accepted in the pass-band?
- How steep is the transition-band?
- How much ripples are accepted in the stop-band?

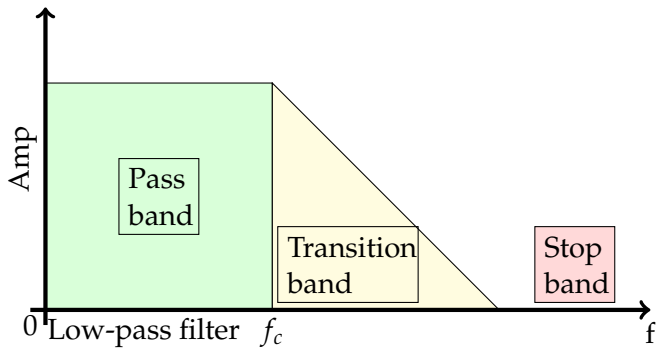


Figure 1.7: Components of filtering

In terms of beamforming, filters that gradually decrease towards the edges of the aperture are being used to suppress the sidelobes, a typical Hamming filter. For maximum penetration we use a rectangular filter in the aperture to gain greater depth. Hours of testing need to be done to be sure that the best filter for the specific application is chosen. This type of testing is time consuming, and a better solution is to use adaptable weights and let the beamformer optimize the weights for you for given criteria. The next step leads us to the following section, where we should take a look on how these weights can be chosen with respect to the recorded data.

1.6 Adaptive Spatial Filter

Adaptive beamformers take advantage of the sensor output, and *adapt* their computations to the observed characteristics. An adaptive method modifies the sensor output weights, and delays the signal according to the observations. Aspects such as signal frequency content, spatial and temporal noise, and signal number are some of the observations taken into account. The weights are custom made to suit the environment, although the environment may change rapidly. Fulfillment of the statistical criteria for optimality is used to find the adaptive weights. When enough observations are available, these kinds of algorithms outperform the DAS beamformer when it comes to performance metrics such as; resolution and signal-to-noise ratio. A disadvantaged is that adaptive algorithms are more sensitive to sensor calibration errors, and that the calculation time is far longer than DAS beamformer.

1.6.1 Minimum Variance Distortionless Response

The *minimum variance distortionless response* (MVDR) beamformer is an adaptive algorithm based on a statistical criterion. The given criterion for MVDR is to minimize the variance while keeping a unit gain in the steering direction, which corresponds to the propagation vector $\vec{d}$.

$$w_{mvdr} = \min_{\vec{w}} \quad \text{VAR}\{z\,[n]\} = \min_{\vec{w}} \quad \vec{w}^H \mathbf{R}_p \vec{w} \tag{1.21}$$

$$\text{subject to } \vec{w}^H \vec{d} = 1 \tag{1.22}$$

The weights can be found by using Lagrange multipliers to solve (1.21) with the constraint (1.22), demonstrated in [30]. The result of the weight calculation is;

$$\vec{w}_{mvdr} = \frac{\mathbf{R}_p^{-1}\vec{d}}{\vec{d}^H \mathbf{R}_p^{-1}\vec{d}} \tag{1.23}$$

The numerator minimizes the minimum variance, while the denominator normalize the response with respect to the propagation vector $\vec{d}$. The calculations are done by assuming zero-mean data vectors, and that we have the true noise spectral matrices $\mathbf{R}_p$.

To illustrate the basic ideas of MVDR we consider a spectral noise matrix $\mathbf{R}_p$ consisting of a single interference signal together with white noise.

$$\mathbf{R}_p = \sigma^2 \mathbf{I} + A_1^2 \vec{v}_1 \vec{v}_1{}^H \tag{1.24}$$

A_1 is the amplitude of the interference with its propagation vector $\vec{v}_1$. σ^2 is the magnitude of the white noise,multiplied with a identity matrix $\mathbf{I}$. By using the matrix inversion lemma we have;

$$\mathbf{R}_p^{-1} = \frac{1}{\sigma^2}\left(\mathbf{I} + \frac{A_1^2}{\sigma^2 + MA_1^2}\vec{v}_1\vec{v}_1{}^H\right) \tag{1.25}$$

Inserting (1.25) into (1.21) will give the following equation

$$w_{mvdr} = \frac{\frac{1}{\sigma^2}\left(\mathbf{I} + \frac{A_1^2}{\sigma^2+MA_1^2}\vec{v}_1\vec{v}_1{}^H\right)\vec{d}}{\vec{d}^H\left(\frac{1}{\sigma^2}\left(\mathbf{I} + \frac{A_1^2}{\sigma^2+MA_1^2}\vec{v}_1\vec{v}_1{}^H\right)\right)\vec{d}} \tag{1.26}$$

By rewriting the equation 1.26

$$w_{mvdr} = \frac{\frac{M}{\sigma^2}\left(\frac{\vec{d}}{M} - \rho\frac{MA_1^2}{\sigma^2+MA_1^2}\cdot\frac{\vec{v}_1}{M}\right)}{\frac{M}{\sigma^2}\left(1 - \frac{MA_1^2}{\sigma^2+MA_1^2}|\rho|^2\right)} \tag{1.27}$$

The resulting weights is generated by two DAS beamformers, one corresponds to the desired signal and the other to the interfering signal. The value ρ is defined as the spatial correlation coefficient between the desired and the interfered signal.

$$\rho = \frac{\vec{v}_1{}^H\vec{d}}{M} \tag{1.28}$$

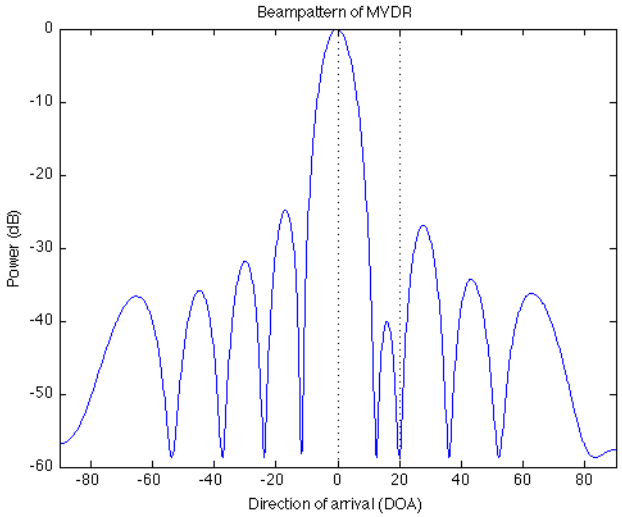


Figure 1.8: Beampatten of MVDR, the beamformer is steered toward 0°,with an interference in 20° degree.

As mentioned ρ indicates how much the respective signals, correlates with each other spatially. If they are fully spatial correlated $\rho = 1$, it will be impossible to separate the interfered and the desired signal, because they overlap in space. Otherwise when the spatially correlation factor ρ has a value $0 \leq \rho < 1$ then the interference is suppressed. When $\rho = 0$ the interference is totally cancelled. The ρ factor combining the desired and interference signal optimally and provide suppression of interference. As illustrated in figure (1.8) we can see that the interfering source arriving from an angle of 20° degrees is totally suppressed.

Minimum Power Distortionless Response (MPDR) is a beamformer with another statistical criteria of optimization, it minimizes the expected power of the beamformers output while keeping a distortionless propagation vector $\vec{d}$:

$$\min_{\vec{w}} \quad \mathrm{E}\{|\vec{z}\,[n]\,|^2\} = \min_{\vec{w}} \quad \vec{w}^H \mathbf{R} \vec{w} \tag{1.29}$$

$$\text{subject to } \vec{w}^H \vec{d} = 1 \tag{1.30}$$

Again the weights are found by using Lagrange multipliers. The difference from MVDR weights is merely that we now uses the full $\mathbf{R} = |A|^2 \vec{d}\vec{d}^H + \mathbf{R}_p$ in the calculations instead of $\mathbf{R}_p$.

$$\vec{w}_{mpdr} = \frac{\mathbf{R}^{-1}\vec{d}}{\vec{d}^H \mathbf{R}^{-1} \vec{d}} \tag{1.31}$$

In an ideal scenario the two beamformer methods are equal, reasoned that the name "MVDR beamformer" is used for both methods. We emphasize that it is only valid for the ideal scenarios, with non-ideal scenarios it has been shown that the MVDR is more robust, because its weights are based on the noise term only. Any wrong assumptions about element

positioning or propagation vector $\vec{d}$ are disregarded in the calculation of the MVDR weights. The MVDR covariance matrix is based on the noise term and exclude everything form the signal and avoids wrong assumption of the signals in the estimates. In both beamformers we are depending on knowing the exact value of $\mathbf{R}$ or $\mathbf{R}_p$, which is often not the case. Since we do not have the true values of either the noise spectral matrix or the data spectral matrix, we estimate $\mathbf{R}$ from the available recorded data. When an estimated covariance matrix is used in the beamformer calculation we call it the *Capon beamformer*, after the work done by J. Capon[6]. The sample covariance matrix $\hat{\mathbf{R}}$ is estimated from the recorded data $\vec{y}\,[n]$:

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{n=0}^{N-1} \vec{y}\,[n]\, \vec{y}\,[n]^H \tag{1.32}$$

, and inserted into equation (1.31) instead of $\mathbf{R}$

$$\vec{w}_{capon} = \frac{\hat{\mathbf{R}}^{-1}\vec{d}}{\vec{d}^H \hat{\mathbf{R}}^{-1}\vec{d}} \tag{1.33}$$

Since the sample covariance matrix is a statitical estimator it requires *enough recorded data* samples to be trustworthy, otherwise the Capon beamformer will perform poorly. The number of time samples has to at least fulfill the criterion $N \geq M$ for $\hat{\mathbf{R}}$ to be invertible. The number of time samples that is recommended is $N > 2M$ [30] to give satisfactory performance from the beamformer. Another issue concerning the Capon beamformer is that it can be sensitive to *model errors* since it is an adaptive beamformer. This may result in severely degraded performance of the beamformer. A critical scenario is if one of the sensors is disabled, and the sensor output gives 0. It is critical because Capon optimize the sensor output by finding the minimum variance, which will be the disable sensor that always gives 0. Model errors can be compensated with robustness techniques that will be discussed in the section of pre-processing (1.7). It is done to ensure we have an inverse matrix $\hat{\mathbf{R}}^{-1}$.

1.6.2 Amplitude and Phase Estimation

The *Amplitude and Phase EStimation* (APES) was first introduced by Li and Stoica [16] and was used outside the beamformer context as the approximate of the *maximum likelihood* (ML). It is now used as the estimate of the complex amplitude (amplitude and phase) of a narrowband signal. The problem of interest was to estimate the amplitude from the recorded data vector, and improve the performance compared with FFT-based amplitude estimation methods. The solution was the APES algorithm. It is an adaptive finite impulse response (FIR) that yields significantly reduction in the sidelobes combined with a higher resolution compared with DAS beamformer. It was derived from a single snapshot,N=1, divided into subarrays under the assumption that the noise vectors are all independently and identically distributed zero-mean Gaussian random vectors. The assumption about independently data vectors is wrong, due

to the overlapping subarrays. So APES algorithm is a complex-amplitude, and a ML estimator only in an approximate sense, as the authors points out in [16], due to dependency of the overlapping vectors. The ML derivation [16] was the first and the original derivation, two other derivations have been made [22, 23] by the same researchers. Let $\vec{y}_k[n]$ denote the spatial smoothing vector related to the K subarrays, and $\vec{w} \in \mathbb{C}^{M\times 1}$ be the vector of filter coefficients. Together they form the filter output $\vec{w}^H\vec{y}[k]$. The deterministic joint problem for APES will then be[4]:

$$(\hat{A}_{apes}, \vec{w}_{apes}) = arg\min_{A,\vec{w}} \frac{1}{K}\left|\vec{w}^H\vec{y}_k[n] - Ae^{j\frac{2\pi}{\lambda}\sin(\theta)\delta_k}\right|^2 \tag{1.34}$$

$$\text{subject to } \vec{w}^H\vec{d} = 1 \tag{1.35}$$

With equation (1.34) the filter output will try to get as close as possible to the sinusoid $Ae^{j\frac{2\pi}{\lambda}\sin(\theta)\delta_k}$. Minimizing 1.34 with respect to A will give:

$$\hat{A}_{apes} = \vec{w}^H\frac{1}{K}\sum_{k=0}^{K-1}\vec{y}_k[n]\,e^{-j\frac{2\pi}{\lambda}\sin(\theta)\delta_k} \tag{1.36}$$

Using the results of the estimated A, and insert it into (1.34) will give the minimization problem:

$$\min_{\vec{w}} \vec{w}^H\mathbf{R}_p\vec{w} \qquad \text{subject to } \vec{w}^H\vec{d} = 1 \tag{1.37}$$

$$\hat{\mathbf{R}}_p = \hat{\mathbf{R}} - \left(\frac{1}{K}\sum_{k=0}^{K-1}\vec{y}_k[n]\,e^{-j\frac{2\pi}{\lambda}\sin(\theta)\delta_k}\right)\left(\frac{1}{K}\sum_{k=0}^{K-1}\vec{y}_k[n]\,e^{-j\frac{2\pi}{\lambda}\sin(\theta)\delta_k}\right)^H \tag{1.38}$$

The optimization problem (1.37) is almost the same as the MVDR beamformer, and the weights are found in the same way (1.22):

$$\vec{w}_{apes} = \frac{\hat{\mathbf{R}}_p^{-1}\vec{d}}{\vec{d}^H\hat{\mathbf{R}}_p^{-1}\vec{d}} \tag{1.39}$$

The APES algorithm finds its weights based on statistics from the interfering signals and noise, by forming a covariance matrix $\hat{\mathbf{R}}_p$ and by subtracting the signal term from the estimated covariance matrix $\hat{\mathbf{R}}$. Since the estimated covariance matrix does not include the desired signal in the optimization problem, by then avoid that the desired signal is being accidentally suppressed. The APES is an accurate and robust beamformer, with better resolution than DAS beamformers, but lower resolution than the Capon beamformer.

1.7 Pre-processing Techniques on the Covariance Matrix

In this section we present techniques that are applied to the sample covariance matrix to improve the estimates, and make the estimates more

[4]We follow the derivation presented in [22]

robust. Adaptive beamformers are sensitive to model errors; therefore we need to apply techniques that will reduce these model errors such as;

- sensors location
- mismatch in sensor calibrations
- speed of propagation
- wrong steering vector
- correlated interfering signals

We know that the MVDR beamformer will, without the distortionless response constraint, force all power to a minimum, which implies that if we have any error in our steering vector or in the model, the consequence could be that it suppresses the signal of interest. The APES beamformer is more robust because it does not include the signal of interest in the optimization. The pre-processing techniques we present here will ensure that an inverse exists, they will decorrelate signal from the interference, and increase the number of data samples.

1.7.1 Diagonal Loading

When calculating array weights for adaptive beamformers an inverse of $\mathbf{R}$, denoted as $\mathbf{R}^{-1}$, is often needed. In terms of linear algebra an $m \times m$ matrix is invertible if it exist a matrix $\mathbf{R}^{-1}$, such that

$$\mathbf{R}\mathbf{R}^{-1} = \mathbf{R}^{-1}\mathbf{R} = \mathbf{I}_M \tag{1.40}$$

where $\mathbf{I}_M$ is an identity matrix of dimension $M \times M$ with ones along the main diagonal. A square matrix that is not invertible is called a singular matrix and can cause serious errors. A singular matrix rarely exist if a randomly matrix is picked it, but to be sure that the randomness never occurs we can force $\mathbf{R}$ to a be non-singular. This is done by *diagonal loading*;

$$\hat{\mathbf{R}} = \mathbf{R} + \delta\mathbf{I}_M \tag{1.41}$$

The diagonal loading factor ,δ, increases the robustness against model errors when we are forming the adaptive weights. By adding δ to the matrix diagonal it introduce some fixed noise/error in the estimation of $\mathbf{R}$. This noise is desirable to be at a minimum, for the case of Capon beamformer with diagonal loading it will converge toward a Delay and Sum beamformer if δ is large [20].

$$\lim_{\delta\to\infty} \frac{(\mathbf{R} + \delta\mathbf{I}_M)^{-1}\vec{d}}{\vec{d}^H(\mathbf{R} + \delta\mathbf{I}_M)^{-1}\vec{d}} = \frac{1}{M}\vec{d} \tag{1.42}$$

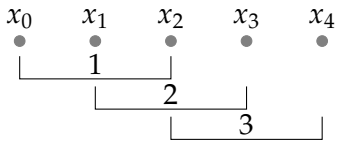
Problems associated with inadequate estimation of the covariance matrix $\mathbf{R}$ can be resolved by diagonal loading, but we have to be aware of desensitizing effect when the loading factor increases. There are many different ways to load the diagonal, but we chose the simplest.

1.7.2 Subarray Averaging

Subarray averaging is used to decorrelate sources [28], we are then able to avoid signal cancellation of the correlated sources. Coherence appears when the phase relation between the desired and the interference signal stays constant. Spatial moving the array can eliminate phase relation; the phase relation becomes then randomized. Moving the array give the effect that each element of the wave phases changes differently according to their direction of arrival. Measuring the cross correlation at several locations makes it possible to gradually nullified the cross correlation terms with subarray averaging [28]. Instead of physically move the array, the cross correlation can be suppressed by spatial averaging. The concept of spatial averaging is to divide the array in K subarrays with the length L. The subarrays are partially overlapping each other, which implies that the number of elements in the array can not exceed the number of a full array ($M > L$). The subarray formulation is presented as

$$z_{sa}(n) = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=0}^{L-1} \vec{w}_k^H \vec{y}_{k+l} \tag{1.43}$$

where M is the number of elements in the array, L is the subarray length and $K = M - L + 1$ is number of subarrays. w_k represents the weighting and y_{k+l} contains the output data. The averaging is illustrated with total elements of $M = 5$, subarray length $L = 3$, resulting in $K = 3$ subarrays.



The advantage of having longer subarray length L, is that close sources can be differentiated. Subarray averaging can be interpreted as a bandpass filter, where a large L gives a narrow band filter, and gives fewer weights. With fewer weights you have less degree of freedom to mitigate the inferring signals and noise. When two sources are properly resolved such that only one source is within the bandpass with length K, then is signal cancellation is avoided, otherwise they may cancel each other. When we are using subarray averaging the size of the correlation matrix are reduced from $M \times M$ to $K \times K$, where K is number of subarrays. The length of the subarray L must be equal or less than number of subarrays K. The correlation matrix of the K'th subarray is denoted as $\hat{\mathbf{R}}_k$

$$\hat{\mathbf{R}}_k = \mathrm{E}\{\vec{y}_k[n]\,\vec{y}_k[n]^H\} \quad (k = 1, \ldots, K) \tag{1.44}$$

$$\hat{\mathbf{R}}_{sa} = \sum_{k=1}^{K} v_k \hat{\mathbf{R}}_k \tag{1.45}$$

where $\vec{v}_k$ is averaging weights for the k'th subarray.

$$\sum_{k=1}^{K} \vec{v}_k = 1 \tag{1.46}$$

Beamformers may take use of subarray averaging, and enhance performance and robustness with regards to signal cancellation. Subarray averaging is also been used to make an estimate of $\mathbf{R}$ when the calculations are based on only a single, or only a few, temporal samples. This technique is used in medical ultrasound imaging where the transmitted pulses are short and non stationary enviroment [25].

1.7.3 Forward-backward Averaging

Forward-backward averaging is a technique used to increase the number of data samples by a factor two and decorrelate the estimates. In the previous section (1.7.2) we introduced the subarray averaging, this was the forward implementation used to decorrelate signals in the estimates. A backward implementation uses the same technique but with the data samples conjugated and flipped [30].

$$\mathbf{R}_{sa\cdot B} = \mathbf{J}\mathbf{R}^{*}_{sa\cdot F}\mathbf{J} \tag{1.47}$$

With the use of the complex conjugate $\{\cdot\}^{*}$ and use of the exchange matrix $\mathbf{J} \in \mathbb{C}^{M\times M}$ the data vectors are flipped:

$$\mathbf{J} = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \dots & 0 & 0 \end{bmatrix} \tag{1.48}$$

To make it possible to flip the matrix and use the conjugate operator without loosing information the array has to be centrosymmetric such as an ULA. The symmetric requirement in the array geometry is the same for every element located in x_m, there must be an identical element at $-x_m$[30]. Many different array structures exists that will satisfy this requirement, but we will concentrate on uniform linear arrays. The new estimate of the covariance matrix with forward-backward averaging is then given by

$$\mathbf{R}_{sa\cdot F/B} = \frac{\mathbf{R}_{sa\cdot F} + \mathbf{R}_{sa\cdot B}}{2} \tag{1.49}$$

1.7.4 Filtering the Covariance Matrix

Filtering the covariance matrix is another approach to decorrelate the coherent signals in the estimated covariance matrix $\hat{\mathbf{R}}$ [15]. The method preforms a linear two-dimensional filter operation, and it does not modify the signals angle of arrival, which is important to maintain when making beamformers. What the filter does modify is the cross-correlation between the signals in the covariance matrix. With the right choice of filter coefficients the signals can be totally decorrelated. The principle of spatial smoothing of the covariance matrix is to divide the matrix into submatrices, which are then filtered. Now the modulated matrix should be free of correlation between the desired and interfering signals if we

have chosen the right filter. Originally we have M sensor elements $\vec{y}[n] = [y_1[n], y_2[n], ..., y_M[n]]$, that form the spatial covariance matrix $\mathbf{R} = \mathrm{E}\{\vec{y}\,[n]\,\vec{y}\,[n]^H\}$, consisting of $M \times M$ relations. The filter kernel is defined as a $Q \times Q$ a complex valued matrix $\mathbf{G}$. Q defines the size of the filter $\mathbf{G}$, and have to be smaller than M ($Q < M$). The operation that preforms the filtering and gives us $\mathbf{R}'$ can be mathematically described as;

$$\mathbf{R}' = \sum_{i=1}^{L}\sum_{j=1}^{L} G_{ij}\hat{\mathbf{R}}_{i+l-1,j+m-1} \tag{1.50}$$

$$l, m = 1, 2, ..., K \tag{1.51}$$

$K = M - L + 1$ describes the dimension of the filtered covariance matrix $\mathbf{R}'$. By doing this type of filtering on the covariance matrix the effective number of array elements are reduced from M to K. This is because $\mathbf{R}'$ is used instead of the original matrix $\hat{\mathbf{R}}$ for the calculations of beamformers. The filter operation described in (1.51) can be formulated as two dimensional convolution

$$\mathbf{R}' = \hat{\mathbf{R}} * \mathbf{G} \tag{1.52}$$

The filter $\mathbf{G}$ can have different size and structure, depending on the intention of the filter. For our purpose the filter $\mathbf{G}$ main intention is to decorrelate the correlation matrix $\hat{\mathbf{R}}$ without loosing the estimates of the arriving waves. To fulfill this criterion five important properties are being made [15];

- The structure after filtering is given by $\mathbf{R}' = \hat{\mathbf{R}} * \mathbf{G} = \mathbf{A}_k\mathbf{S}'\mathbf{A}_k^* + \mathbf{W} * \mathbf{G}$, where $\mathbf{A}_k$ is a $K \times X$ matrix and X describes the number of coherent waves. Preservation of the signal information is important such that the new estimate of $\hat{\mathbf{R}}$ can be used for calculation of the adaptive beamformers.

- The filtered signal component $\mathbf{S}'$ is related to the original signal correlation matrix S by the equation $\mathbf{S}' = [\mathbf{A}_L^T\mathbf{G}\mathbf{A}_L^*] \circ \mathbf{S}$. $\mathbf{A}_L$ is a $L \times X$ matrix whose columns are $L \times 1$ arrival of array vectors. X is the number of signals with identical center frequencies impinge upon the array. $\mathbf{S}$ is then element-wise factorized with $[\mathbf{A}_L^T\mathbf{G}\mathbf{A}_L^*]$ to achieve $\mathbf{S}'$.

- The filter $\mathbf{G}$ must satisfy the following statement to be able to decorrelate X coherent sources completely. $\mathbf{A}_L^T\mathbf{G}\mathbf{A}_L^* = \mathbf{I}_X$. $\mathbf{I}_X$ defines the identity matrix of size X. With this statement satisfied we are guaranteed that $\mathbf{S}'$ is a diagonal matrix and the signal information is preserved in $\mathbf{S}'$.

- By consider that the noise correlation matrix $\mathbf{W}$ as a Toeplitz matrix, then $\mathbf{W} * \mathbf{G}$ also is a Toeplitz matrix for any filter $\mathbf{G}$. The filter $\mathbf{G}$ have two useful and interesting constraints, first $\mathbf{G} = \mathbf{G}^H$ states that the filter is a Hermitian matrix, and second the $Trace(G, q) = \delta_q \quad q = 0, 1, ..., Q - 1$ states that the sum of of element across the main diagonal is 1 and 0.

- $\mathbf{R}'$ is positive definite as long as $\mathbf{G}$ is chosen in a prescribed way and the dimension of $\mathbf{G}$, L , exceeds a threshold.

Describing spatial smoothening with this filter approach is a straight forward task. The weights on the diagonal shall be fixed, and each element shall be equally weighted. The filter is made of an identity matrix $\mathbf{I}$ with the filter size Q, then the matrix is factorized by the averaging coefficient $\frac{1}{Q}$. This is how the approach of spatial smoothing with subarrays approach can be interpreted on a two dimension filter term, with the filter $\mathbf{G} = \frac{1}{Q}\mathbf{I}_Q$. Instead of having fixed weights on the main diagonal of $\mathbf{G} = diag(g_1, g_2, ..., g_Q)$, each parameter g_q can be chosen differently and are adjustable.

$$\mathbf{R}' = \sum_{q=1}^{Q} g_q \hat{\mathbf{R}}_q \tag{1.53}$$

$\mathbf{R}_q$ is a cutout from the original covariance matrix $\mathbf{R}$. This type of filter matrices are used in optimization problems, such as adaptive spatial averaging that we will look closer into in a later chapter.

1.8 Beamformer Application

Many similarities exists in the different types of beamforming applications, the key difference are the operating frequency on the transmitted signal and the speed variation of the propagation due to physical conditions. The theories behind the active systems have a lot in common,whether its space,earth or ocean the application operates. The fundamental difference is that the propagation of acoustics energy in ocean/body is significantly more complicated than the propagation of electromagnetic energy in the atmosphere. These propagation characteristics have a major influence in the design of acoustics waves for the ocean/body [30]. In recent decades adaptive beamformer weights has been applied to the field of array processing such as wireless communication, radar, sonar and medical ultrasound imaging [25, 17, 16]. In this section we will give an insight in the application of medical ultrasound imaging with the use of adaptive weights.

1.8.1 Medical Ultrasound Imaging

Medical imaging is a technique used to create images of the internal body, which are studied to detect abnormalities inside the body without medically surgery. The images are created by transmitting high frequency sound or ultrasound waves into the body followed by reception, processing, and interpreting the returning echos from structures and tissues. Ultrasound is sound waves with frequencies greater than the upper limit of human hearing ($\sim 20kHz <$), with its purpose to penetrate the body and measure the reflected signal. The reflection signatures reveal details about the inner structure of the body, which may have high medical interest. The medical ultrasound system relies on the fact the biological tissues scatters or reflects

incident sound. Scattering refers to interaction of between the transmitted sound wave and particles that is smaller than the sound's wavelength λ, while reflection is the echos from particles larger than λ [1]. The propagation characteristics of a material are related to its sound speed, significant discontinuities in the characteristics of a material will reflected or scatter the sound strongly. In medical ultrasound a probe is used to transmit and record the ultrasound echos. The probe consists of an array that generates a focused pulse that is being transmitted into some portion of the body. The pulse will propagate in the body and is reflected back to the probe by inhomogeneities in the tissue. These reflections are analyzed with respect to strengths of the reflection and the propagation time. The strength of the echo is used to estimate density of the medium, and with the propagation time we can locate the location. When arranging the estimates in time we got an image line, and by repeating this process by sending out focused pulses in adjacent direction will complete the sector image.

Specialties in medical ultrasound imaging are that we can always operate in the near field. The speed of the propagation c is significantly lower than the speed of light and the multi-path propagation phenomena is usually a problem. There exist many similarities between medical ultrasound imaging and sonar imaging, because of the operating environment has many of the same effects.

1.8.2 Related Work

In the last couple of years it has been an increased focused on applying adaptive beamformers to medical ultrasound systems. The increased focus has resulted in higher resolution and better contrast in the medically images. To apply the adaptive beamformers to the application medical ultrasound imaging, the methods have been simplified and more compact to fit the hardware limitations. The most common approach are based on minimum variance, which was derived by Capon[6].

The improvements of the adaptive beamformers come at expenses of increased computational complexity, which makes the adaptive beamformers less attractive in spite of their performance advantaged. Low-complexity adaptive beamformers such as Synnevag *et al.* [26] have been successively applied to medically ultrasound systems to overcome the hardware limitations. They use several predefined windows and use the minimum variance optimization criterion to decide which window to use at a specific point in the image. Nilsen and Hafizovic [19] have investigated the application of a beamspace adaptive beamformer for medical ultrasound imaging, which can be used to achieve reduced complexity with performance comparable to that of the Capon beamformer. Other methods have been proposed to reduce the complexity, such as reducing the number of elements, hence directly reducing the complexity[31].

Other issues to apply adaptive beamformers to medical ultrasound is that the adaptivity is highly sensitive to errors in the steering vector, and the observations are naturally correlated because the backscattered echoes is composed of replicas of the same transmitted waveform. The data has

to be decorrelated to avoid effects as signal cancellation of the desired signal. The causes of steering vector errors are usually incorrect sound estimate. The speed estimates varies with the different characteristics of the tissue in the human body, and to deal with these types of errors robustness methods such as variants of diagonal loading, spatial averaging, and forward-backward estimation has been suggested [25, 10, 2]. Wang *et al.* [33] has contributed with a synthetic aperture approach together with a robust Capon beamformer and showed increased resolution and contrast in the context of medical ultrasound imaging.

Viola and Walker [32] presented a series of adaptive beamformers in the field of medical ultrasound imaging and compared their performance to data independent beamforming techniques.

The conducted studies have given broader knowledge and understanding of the varieties in the medical imaging application, which again has lead to more advanced beamformer designs that more suitable for the medical ultrasound purpose.

Part II

The Project

Chapter 2

Planning the Project

2.1 Motivation

When making beamformer algorithms that are based on estimates of the covariance matrix $\hat{\mathbf{R}}$, it is fundamental to get a good estimate. A good estimate is close to the true $\mathbf{R}$, and should not introduce new artifacts or errors. The aim is to make an estimate that is as close as possible to the true value preferably without significantly increasing the computational cost. Several factors influence the result and therefore why a perfect $\hat{\mathbf{R}}$ is not obtained, such as destructive errors and model deviations. Destructive errors occur in the model, as model deviations. Typically model deviations are physical errors such as wrong sensor position or mismatched sensors. Errors that occur in the field are also destructive for the estimation; field errors can be coherent signals that arise due to multi-path propagation or when "smart" jammers deliberately induce coherent interference. The estimates can also fail due to a finite number of samples; with only a few samples the underlying statistics will return wrong parameters, the estimates are then based on these wrong parameters which result in an unsatisfactory $\hat{\mathbf{R}}$ matrix.

To emphasize the importance of a good estimate, the covariance matrix $\hat{\mathbf{R}}$ is used in many optimization problems and poor estimates will causes errors and inaccuracy. So if the estimates are insufficient, all other products of the estimate will be bad too, and the beamformer will probably not succeed if it is based on a poorly estimated covariance matrix. It is especially critical for the case of the Capon, which is an aggressive beamformer method that tries to force everything to zero except for the constraint. For this method an inaccurate covariance matrix may result in fatal errors, the method can force wrong angle of arrival to a minimum due to a wrong estimate of the desired signal.

This begs the question: What is the actual true value of $\mathbf{R}$? In general we do not have any kind of prior knowledge about the field we are trying to sense with the array. The knowledge that might have helped, is the numbers of interfering sources with their directions of arrival. But this information is not available in the general case; the only thing we can be sure about is the intended array structure or some

superficial knowledge about the nature of the noise. In many practical applications this information is known, which can and should be taken into consideration when determining the formulation of the estimation of statistics. For our case we have chosen to use a uniform linear array, which for a perfect model will result in a covariance matrix that has a Toeplitz structure, which is constant along the diagonals. The Toeplitz matrix model the covariance of a random vector obtained by sampling a spatially stationary wave field with a uniform linear array. Since this is the only definite knowledge we have about the true covariance matrix we suggest that it should be taken into consideration when we are doing estimations of our covariance matrix. To show how the true **R** would look like for a simple example with one desired source, one interfering source and with stationary noise. We then have the following equation for **R**;

$$\mathbf{R} = \mathrm{E}\{\vec{y}\,[n]\,\vec{y}\,[n]^H\} = |A_d|^2\vec{v}_d\vec{v}_d^H + |A_i|^2\vec{v}_i\vec{v}_i^H + \sigma_n^2\mathbf{I} \tag{2.1}$$

The power of the signal is described by the power of the amplitude A,and $\vec{v}$ is the propagation vector. The noise term is described by the noise variance σ_n^2 , and multiplied by the identity matrix which spans the entire space. Since each of the terms in Eq. (2.1) are on a Toeplitz form, the sum of them will also have a Toeplitz structure. Exploitation of Toeplitz structured covariance matrices was demonstrated by Gray [9] and Fuhrmann [7]. The constrained Toeplitz structured covariance matrix were researched early by Burg, Luenberger and Wenger[5], and afterwards by Miller and Snyder [18], and Tourtier and Scharf [29].

An advantage of working with Toeplitz matrices is that we always can be sure that an inverse exists. With the structure of a Toeplitz matrix we can reduce the arithmetic operations that is need to invert the matrix, which again reduces the inverting time. It may not reduce the total time since it may cost some time to convert the matrix into a Toeplitz matrix. The representation and storage of the matrix can be reduced to one or two vectors, depending on if it is symmetric or not, but this effect is not important in our case.

The effect of forcing the estimate to a Toeplitz structure will be investigated through testing beamformers with metrics that are essential for a beamformer. Would the Toeplified matrix be closer to the real **R**? With a Toeplitz covariance matrix would the Capon performance increase? What is the best method to make a Toeplitz matrix without loosing fundamental information? The purpose of this thesis is to examine the possible performance improvements obtained by incorporating Toeplitz estimation into adaptive beamformers.

2.1.1 Toeplitz

For adaptive beamformers, based on output minimization, performance is best when the spatial covariance matrix is diagonal, due to avoided signal cancellation of the desired signal. Such diagonal matrix is a Toeplitz matrix where all diagonals are constants, which is to say that any $M \times M$ matrix

A on the form

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \dots & \dots & a_{-M+1} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{M-1} & \dots & \dots & a_2 & a_1 & a_0 \end{bmatrix} \tag{2.2}$$

is a Toeplitz matrix. Exploiting the Toeplitz structure has shown that these techniques has led to a significant performance improvement for beamformers that require the inverse of the estimates $\hat{\mathbf{R}}^{-1}$ [7]. The variety of problems solving equations containing the Toeplitz matrix, has led to many efficient algorithms for solving these equations. The structure of a Toeplitz matrix makes it possible to represent the matrix with two vectors if it is symmetric. Systems of $\mathbf{A}x = b$ can be solved in $O(M^2)$ arithmetic operations, with Levinson-Durbin algorithm. Another positive effect of working with a Toeplitz matrix; is that the inverse always exist. If the Toepliz matrix in addition is a circulant matrix the equation can be solved quickly with fast Fourier transform (fft), and requires only $O(M\log(M))$ operations.

2.1.2 The Frobenius Norm

The Frobenius norm is a matrix norm that is defined as the square root of the absolute sum of every element squared;

$$|A|_F = \sqrt{\sum_m \sum_n |a_{m,n}|^2} \tag{2.3}$$

This norm can be used as a measurement of how much the estimated covariance matrix $\hat{\mathbf{R}}$ deviates from the true $\mathbf{R}$ by finding the total deviation across all matrix elements.

$$|\mathbf{R} - \hat{\mathbf{R}}|_F = \sqrt{\sum_m \sum_n |r_{m,n} - \hat{r}_{m,n}|^2} \tag{2.4}$$

The measurement can be used as an indication of how good the estimates are, or possibly give us a threshold how much the estimates can deviate before it becomes useless.

2.2 Toepliz structured covariance matrix

In this section we will describe the methods that turn our estimates of the covariance matrix into Toeplitz structured matrices.

2.2.1 Spatial Convolution

Synnevåg and Jensen suggested the spatial convolution method in [27], which will be described in detail in this section. Assume a uniform linear array and a single snapshot ($N = 1$), with a single snapshot of the wave field the time dimension is discarded. Instead we introduce a location dependence to derive an alternative approach for the optimization. Expressing the single snapshot of the sensor outputs can be written as;

$$z_0 = \sum_{m=0}^{M-1} w_m^H y_m \tag{2.5}$$

This summation (2.5) can be interpreted as a windowed convolution of the weights w_{-m} and the sensor outputs y_m. Using the Fourier transform to convert into a function of wavenumber $k = \frac{2\pi}{\lambda}\sin(\theta)$ that describe the angle of arrival. One of the Fourier property is stated as

$$w[m] * y[m] \overset{\mathcal{F}}{\Longleftrightarrow} W(k)Y(k) \tag{2.6}$$

where W(k) and Y(k) are the discrete-time Fourier transforms of w[m] and y[m]. Using Parseval's relation on our single snapshot (2.5) gives us;

$$z_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} W^*(k)Y(k)dk \tag{2.7}$$

The equation integrates the sample space from $-\pi \leq k \leq \pi$. The power estimate or the variance of z_0 can then be consider as;

$$|z_0|^2 = \left|\frac{1}{2\pi}\int_{-\pi}^{\pi} W^*(k)Y(k)dk\right|^2 \tag{2.8}$$

From this equation the optimization with the constraint yields

$$\begin{aligned} \min_{w} \quad & \left|\frac{1}{2\pi}\int_{-\pi}^{\pi} W^*(k)Y(k)dk\right|^2 \\ \text{subject to} \quad & W^*(k_d) = 1 \end{aligned} \tag{2.9}$$

The constraint ensures that we have unit gain in the look direction k_d. The problem with this equation is that if there is more than one source in the wave field, it can lead to signal cancellation. The signal cancellation takes place when an unwanted signal has a solution $W^*(k_1)Y(k_1)$, which has opposite sign of the desired direction $W^*(k_d)Y(k_d)$, and then cancels the signal of interest although the constraint is satisfied. This problem is well known for optimization beamformer such as Capon, when the signals are coherent[34]. The problem herein is that the optimization criterion stated in (2.9) enables the product $W^*(k)Y(k)$ to have both negative and positive contributions, and therefore lead to signal cancellation. To avoid the problem we can ensure only positive contribution to the cost function. Instead of using the optimization (2.9) criterion we rather use;

$$\begin{aligned} \min_{w} \quad & \frac{1}{2\pi}\int_{-\pi}^{\pi} |W^*(k)Y(k)dk|^2 \\ \text{subject to} \quad & W^*(k_d) = 1 \end{aligned} \tag{2.10}$$

The square is moved inside the integral and assures a positive contribution for every wavenumber k. The remaining part is to convert the optimization criterion back to the spatial domain by using the Fourier transform (2.6). The optimization criterion in the spatial domain after the Fourier transform becomes;

$$\begin{aligned} \min_{w} \quad & \sum_{m=-M+1}^{M-1} |\vec{w}_m^H \vec{y}_m|^2 \\ \text{subject to} \quad & \vec{w}^H \vec{d} = 1 \end{aligned} \tag{2.11}$$

We now have a minimization that deviates from the original approach (1.31);

$$\begin{aligned} \min_{w} \quad & \vec{w}^H \hat{\mathbf{R}}_{sc} \vec{w} \\ \text{subject to} \quad & \vec{w}^H \vec{d} = 1 \end{aligned} \tag{2.12}$$

To solve this minimization we have to construct the *spatial convolution* estimated covariance matrix $\hat{\mathbf{R}}_{sc}$. The matrix $\hat{\mathbf{R}}_{sc}$ is constructed by the original sensor outputs that are padded by $M-1$ zeroes on both sides of the data sequence $\vec{y}[n]$, resulting in a total length of the zero-padded vector of $3M-2$.

$$\hat{\vec{y}}[n] = [0, 0, \ldots, 0, y_0[n], y_1[n], \ldots, y_{M-1}[n], 0, \ldots, 0, 0]^T \tag{2.13}$$

The vector $\hat{\vec{y}}[n]$ is then divided into subarrays of length M to calculate the spatial convolution matrix using subarray averaging. The zero-padded vector ,$\hat{\vec{y}}[n]$, is only used for the calculations of the weights, provided by forming the spatial covariance matrix. However, after this calculations the weights are applied to the original data vector $\vec{y}[n]$. The difference between the standard subarray averaging method and spatial convolution is how every single sensor output is involved. With the spatial convolution method every single sensor output contributes equally to the covariance matrix. With the standard subarray averaging the sensors at the edges of the array contributes only one time in the averaging, and the sensors in the middle of the array contributes more to the averaging.

2.2.2 IAA - APES

The iterative adaptive approach is used in amplitude and phase estimation (IAA- APES). It is a data-dependent,non-parametric algorithm based on weighted least squares approach, and it resembles from the APES algorithm [16, 22, 23]. It is a non-parametric algorithm that attempts to compute the signal estimate without using a priori information or making any explicit assumption about the signal model. This result tends to be more robust than the parametric alternatives. The construction of the covariance matrix $\hat{\mathbf{R}}_p$ in IAA-APES is made iterative, and differs from the original approach of APES. In the original approach the APES covariance matrix is based on subarray averaging, and is constructed by subtracting

the desired signal from the estimates of the covariance matrix, so that only the noise and inferring signal are remaining in the estimates. This ensures that we do not make any wrong assumption of the signal of interest, which can cause poor performance for the beamformer. The number of signals is usually unknown so we make it more robust by making a grid of interest. Let G be the number of scanning points/angles in the region. The IAA-APES method makes an overkill assumption about the number of interfering signals in the region by interpretation of every scan-line as a potential interfering source. When representing the grid the IAA-APES makes an angle matrix $\mathbf{A} = [\vec{a}(\theta_0), \vec{a}(\theta_1), \ldots, \vec{a}(\theta_G)]$ where each column vector $\vec{a}(\theta_g)$ represent one scan-line in the grid. To construct the estimated covariance matrix $\hat{\mathbf{R}}_{apes}$ IAA-APES uses the signal power $\hat{P}$ as weights;

$$\hat{\mathbf{R}}_{apes} = \sum_{g=0}^{G-1} \hat{P}_g a(\theta_g) a(\theta_g)^H = \mathbf{A}\text{diag}(\hat{P}_{prev})\mathbf{A}^H \tag{2.14}$$

The function *diag* places the power estimate along the main diagonal and else where there is zeros, the resulting matrix has a dimension of $G \times G$. As mentioned, IAA-APES is an iterative method, so the previous signal power $\hat{P}_{prev}$ is used to obtain the current signal power. Since the algorithm is iterative it implies that we require an initialization. The initialization is done by a standard delay-and-sum beamformer method. The power estimate $\hat{P}$ is used to make a diagonal matrix. It contains the power at each angle of the scanning grid and the number of scan lines determines the dimension. The diagonal structure $\text{diag}(\hat{P}_{prev})$ is used to update $\hat{\mathbf{R}}_{apes}$ and makes the algorithm work properly even for low numbers of time samples and coherent sources. The resulting matrix has a *Toeplitz structure*, because every $P_g a(\theta_g) a(\theta_g)^H$ are Toeplitz will the sum be Toeplitz as well. When using the signal power estimate as weights we are able to detect every interfering source and the adaptivity will be used to suppress the inference. An advantage with IAA-APES method is that the beamformer can be applied to arbitrary array geometries, due to the iterative solution. The method is summarized in table (2.1).

The IAA-APES methods does not provide significant improvements in performance after about only 15 iterations [36], this is illustrated in figure (2.1) where we steer towards 0° together with an interfering source in 40°, and a SNR value at 30 dB.

2.2.3 Adaptive Spatial Averaging of $\hat{\mathbf{R}}$

We use the method subarray averaging to decorrelate the coherent signals in the estimation of the covariance matrix $\hat{\mathbf{R}}$, but with subarray averaging such as we introduced in section (1.7.2) the subarrays are uniformly combined in Eq. (1.46). We now want to implement this method and start with the same basis as for regular subarray averaging. The *adaptive spatial averaging* method that is suggested in [28] does not use uniformly

Table 2.1: Pseudocode of IAA-APES

% Initialization by DAS
for all θ **do**
$\hat{P} = \frac{1}{N}\sum_{n=1}^{N} |\vec{w}^H \vec{y}\,[n]\,|^2$
end for
% IAA-APES convergence
while convergence **do**
$\hat{\mathbf{R}}_{apes} = A(\theta)\mathrm{diag}(\hat{P})A(\theta)^H$
for all θ **do**
$s_g(n) = \frac{a(\theta_g)^H \hat{\mathbf{R}}_{apes}^{-1} \vec{y}[n]}{a(\theta_g)^H \hat{\mathbf{R}}_{apes}^{-1} a(\theta_g)} \qquad n = 1, \cdots, N$
$\hat{P}_g = \frac{1}{N}\sum_{n=1}^{N} |s_g(n)|^2$
end for
end while

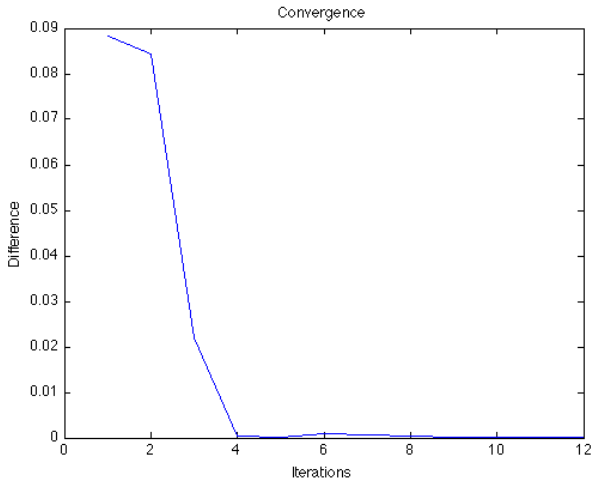


Figure 2.1: Convergence of IAA-APES method, shows the difference between previous and current power estimate, P.

distributed weights, but it chooses adaptive weights for v_k.

$$\hat{\mathbf{R}}_{sa} = \sum_{k=1}^{K} v_k \hat{\mathbf{R}}_k \tag{2.15}$$

The weights v_k are selected based on the knowledge that if we have a covariance matrix without cross correlations between the desired and the interference signals, we should have a $\mathbf{R}$ matrix with a Toeplitz structure. This is only valid under the assumption of identical array elements. With only subarray averaging over the element inputs and uniformly weights, the interference signal is not being suppressed completely. This is due to the constraints of the covariance matrix, which cannot be made a Toeplitz matrix with only simple averaging. Since we know it should be a Toeplitz matrix we can therefore force it to become the wanted form by using adaptive weights v_k. These weights can equalize the

elements on the diagonals so that the matrix can be close to Toeplitz. To determine the weights we start by defining a quantity ϵ, this quantity defines the deviation between our estimated matrix and a Toeplitz matrix. As mentioned earlier each element along the diagonal in a Toeplitz matrix has equal constant value. To fulfill this statement we do an averaging over each diagonal and use it to equalize the diagonal. The deviation quantity ϵ is defined as

$$\epsilon = \sum_{i=0}^{L-2} \sum_{l=1}^{L-i} |\hat{r}_{l+i,l} - \hat{r}(i)|^2 \tag{2.16}$$

The averaging over the i'th diagonal is described with $\hat{r}(i)$ and $r_{l+i,l}$ is one specific element in $\mathbf{R}$ on the i'th diagonal. The Eq. (2.16) can be written on a more compact linear algebra form [28];

$$\epsilon = \vec{v}^T \mathbf{R}_{ee} \vec{v} \tag{2.17}$$

where

$$\vec{v} = [v_1, v_2, ..., v_K]^T \tag{2.18}$$

$$\mathbf{R}_{ee} = \sum_{i=0}^{L-2} \sum_{l=1}^{L-i} Re\left\{ e_{l+i,l} \cdot e_{l+i,l}^H \right\} \tag{2.19}$$

$$e_{l+i,l} = [e_{l+i,l\cdot 1}, ..., e_{l+i,l\cdot K}]^T \tag{2.20}$$

$$\begin{aligned} e_{l+i,l\cdot k} &= r_{l+i,l\cdot k} - \frac{1}{L-i} \sum_{l=1}^{L-i} r_{l+i,l\cdot k} \\ (i &= 0, ..., L-2; l = 1, ..., L-i; k = 1, .., K) \end{aligned}$$

$Re\{\cdot\}$denotes the real part and the number of interest. $e_{l+i,l\cdot k}$ is the deviation from the element value at $(l+i, l)$ in $\mathbf{R}_{sa\cdot k}$ and the average on the i'th diagonal. This is a standard optimization problem, and our task is to minimize the following formulation;

$$\min_{\vec{v}} (\epsilon = \vec{v}^T \mathbf{R}_{ee} \vec{v}) \tag{2.21}$$

$$\text{subject to } \vec{v}^T \vec{1} = 1 \tag{2.22}$$

The unity vector $\vec{1}$ consists of only of ones and has the length K. This form for minimization is the same as we used when we found the weights for the Capon beamformer in (1.21). The optimum value of $\vec{v}$ is then determined and is given as;

$$\vec{v} = \frac{\mathbf{R}_{ee}^{-1} \vec{1}}{\vec{1}^T \mathbf{R}_{ee}^{-1} \vec{1}} \tag{2.23}$$

These optimal weights are determined given the assumption that it exist an inverse of $\mathbf{R}_{ee}$. The $\mathbf{R}_{sa}$ with adaptive weights can be used for DAS beamformers as well as Capon beamformer. It is only the covariance matrix that is modified in the different cases, a pre-processing method is used in the modification and can by applied to several beamformers. When reducing the size of the covariance matrix from $M \times M$ to $K \times K$, we also

reduce the degrees of freedom. Less degrees of freedom result in fewer adaptive weights that can suppress unwanted terms. This is a limitation that manifests itself for adaptive spatial averaging, this restricts how many coherent signals we are be able to suppress. The size of the full array is sacrificed in order to decorrelate the incoming waves. The degrees of freedom for a subarray averaged matrix $\mathbf{R}_{sa\cdot k}$ is related to each independent term in $\mathbf{R}_{ee}$. The degrees of freedom is calculated by the number of terms corresponding to the length of the main diagonal of $\mathbf{R}_{sa\cdot k}$, minus one is preserved as a reference $\hat{r}(0)$. The number of degrees of freedom for each subarray will then be $L-1$. So assume that we have the desired signal and J coherent signals we have then $J+1$ signals incident on the array. Each element in the covariance matrix will consist of $J+1$ autocorrelation terms that should be preserved, and the $J(J+1)/2$ cross correlation should be suppressed. A simplified example is when we assume that we have one desired and one inference signal, and the noise is neglected. The total array output can be expressed in terms of the desired signal s and interference i.

$$\vec{y} = \vec{s} + \vec{i} \tag{2.24}$$

R is expressed as follows:

$$\mathbf{R} = E\{\vec{y}\vec{y}^H\} = (\vec{s}+\vec{i})(\vec{s}+\vec{i})^H = \overbrace{\vec{s}\vec{s}^H + \vec{i}\vec{i}^H}^{\text{auto correlation}} + \overbrace{2\vec{s}\vec{i}}^{\text{cross correlation}} \tag{2.25}$$

For this example we will have two auto correlation terms, and one cross correlation term. We then consider the cross correlation terms as complex numbers, so that the numbers to suppress is $J(J+1)$. The total degrees of freedom for a $\hat{\mathbf{R}}_{sa\cdot k}$ is then $(L-1)^2$, which gives us the inequality;

$$(L-1)^2 \geq J(J+1) \tag{2.26}$$

Solving the equation with the knowledge that L and J are positive integers, it can be simplified to

$$L \geq J+2 \tag{2.27}$$

This inequality is valid for each subarray with a size of L. The next step that is considered is the number of subarrays. The available degrees of freedom are the number of weights K minus one, it is important that this is used as reference or constraint. Resulting in $K-1$ weights, and a still complex cross correlation that shall be suppressed and eventually vanished. The relation between K and J is expressed as;

$$\begin{aligned} K-1 &\geq J(J+1) \\ K &\geq J(J+1)+1 \end{aligned} \tag{2.28}$$

For one coherent interference signal the size of the subarray must be $L \geq 3$, and we must have $K \geq 3$ subarrays.

2.3 Performance Metrics for Beamformers

To be able to compare our different Toeplifeid beamformers we have chosen to investigate different metrics that should be of importance when choosing the beamformers. The requirements associated with the application will determine which beamformer that should be applied. There will be some beamformers that perform very well with respect to some metrics but is not chosen because they do not perform adequate with respect to other metrics. We have chosen to study the following metrics in greater detail;

- Signal to Noise Ratio (SNR) / Root Mean Square Error (RMSE)
- Array gain
- Error in Direction of Arrival
- Resolution
- Numerical complexity

2.3.1 Signal to Noise Ratio and Root Mean Square Error

Since we working with beamformers with a constraint to keep the unit gain in the steering direction, the signal-to-noise ratio and root mean square error will be the inverse of each other. Both metrics are taken under consideration since they will give different results for beamformers that are constrained different.

Signal to Noise to Ratio (SNR) is a metric that compares the level of desired signal to the level of background noise. The definition is stated as the ratio between signal power and noise power. Given by;

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{2.29}$$

It is desirable to have a significant higher signal power than noise power to be able to extract the information. A ratio higher than 1 indicates more signal than noise, and in many cases we want to maximize this ratio. Low signal-to- noise ratios are among the many challenges that array processing system frequently face.

Root Mean Square Error (RMSE) is a good measurement of accuracy, and it gives us the differences between the true value and the estimated or received value. RMSE is a kind of generalized standard deviation. RMSE is given as;

$$RMSE = \sqrt{|A - \hat{A}|^2} \tag{2.30}$$

where A is the actual value amplitude and $\hat{A}$ is estimated amplitude.

2.3.2 Array Gain

The array gain is defined as the ratio between the SNR beamformer output, and the SNR from one sensor from the array. The metric reflects the improvement in SNR obtained by using a beamformer. It is defined as;

$$AG = \frac{SNR_{out}}{SNR_{sensor}} \tag{2.31}$$

An array gain between $0 < AG \leq 1$ indicates that the beamformer has decreased in performance with respect to SNR. It is desirable to have the highest possible value of AG, which shows the improvement done by the beamformer. For an array consisting of M elements in a field of spatially white noise the array gain will be proportional to the number of elements M.

2.3.3 Error in Direction of Arrival (DOA)

To estimate the directions of arrival, the region of interest are scanned and the point with highest value also called the peak, is estimated as the direction of arrival. When estimating the direction of arrival the result can be imprecise because the peak deviates from the actual direction, it is desirable to know how much the estimate deviates from the actual direction of arrival. The estimated direction of arrival can deviate significantly from the true direction when the covariance matrix is poorly estimated. A result of this type of error can be significantly worse when using Capon beamformer, which then turns out to be more unfavorable than the standard DAS beamformer [17]. In applications where the direction of a source is critical it is important to be aware of the errors in the DOA estimates. Applications can operate with an error threshold that specify the acceptable error in the direction estimates. To measure this metric we have used the knowledge of the actual position to the source and compared it with the estimated value.

2.3.4 Resolution

Resolution is the one of the most common metrics used when comparing beamformers. Resolution is a measurement on how close to sources can be before it is impossible to separate them from each other. Two sources are separated if the distance between the two is bigger than the resolution threshold Δr.

$$|s_2 - s_1| \geq \Delta r \tag{2.32}$$

The resolution of the DAS beamformer is defined by the width of the main-lobe, which can be measured by two different methods. One of the methods to measure the width is to calculate the difference on the x-axis from the peak of the mainlobe to the first zero crossing point. The latter method is to start on the top of the main-lobe and move down the main-lobe until the difference between the top and the current point is a predefined threshold value. Usually,the threshold is estimated to be -3 dB. How the

resolution should by measured is application and system dependent. A beamformer with poor resolution is not able to detect small artifacts in the environment. For some applications, the beamformer is used to make images, the beamformer resolution is therefore crucial for the resolution of the final product. An illustration of the concept of resolution regarding beamformers a steered response is made were the blue scenario has not resolved the two source, while the red case has managed the to separate the two sources in -5° and 5° degrees. The reason that the red case managed to separate the two sources, is because its array has twice the size of the blue case. With a larger array the amount of data increases, which leads to more data to process. To measure the resolution for our beamformers we

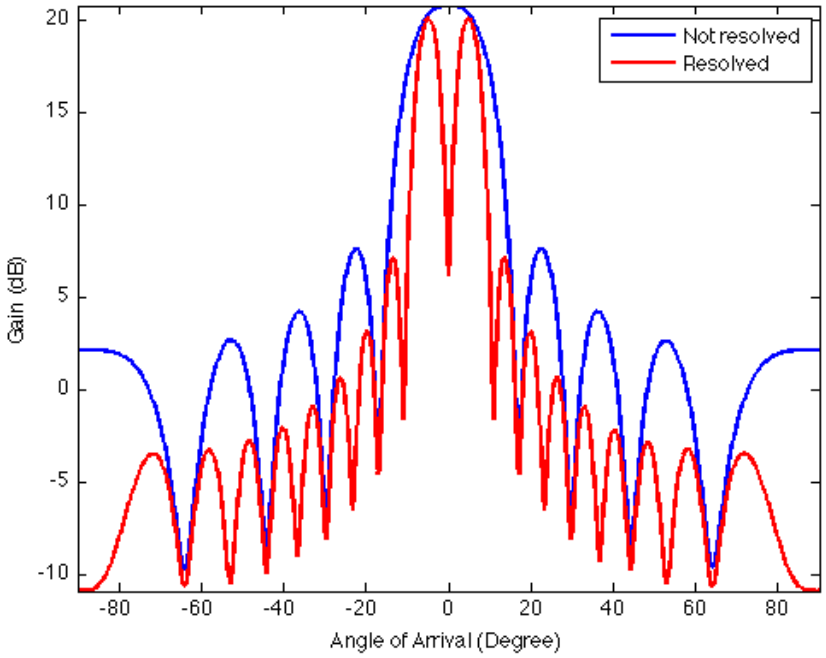


Figure 2.2: Resolution example

have used the -3 dB threshold, by finding the minimum spacing between two sources that gives a -3 dB dip between the sources.

2.3.5 Numerical Complexity

Numerical complexity determines which beamformer that can be used for which application when time and hardware resources represent critical factors. If the application shall serve close to real-time then the beamformer algorithm has to be fast. We chose to measure this metric with how many arithmetic operations are required for the specific beamforming algorithm. We analyze each beamformer with the standard Big O-notation, which tell us the dependence of the complexity on the size of the data. The analysis covers the required costs related to constructing a Toeplified structure and then inverting the matrix, versus the inversion alone. A distinct advantage

of having a symmetric positive definite Toeplitz-structured matrix $\hat{\mathbf{R}}$ with size $M \times M$ is that its inverse can be calculated in $O(M^2)$ [8], while standard Gaussian and Cholesky factorization techniques require $O(M^3)$ arithmetic operations. The fast $O(M^2)$ algorithm is the Levinson-Durbin recursion which was first introduced by Norman Levinson in 1947, later improved by J. Durbin in 1960, and subsequently by W.F Trench and S. Zohar. Since we investigate the Toeplitz constrained covariance matrices, we will benefit from the short inversion time. But as mention earlier, the cost of the constraint for the estimate must be calculated.

Part III

Results and Discussion

Chapter 3

Results

3.1 Problem Formulation

We have made a general model that covers several scenarios for different beamformer applications. Many of the scenarios will have adjustable parameters when applied in computer simulation. Since the general field is a random field, we do not have any kind of information about how many signals that will propagate or the level of noise that will be in the field. Other effects that affect the performance of the beamformer may be present in the field, such as coherence. Errors in the physical setup of the model, typically wrong positioning of a sensor or uncalibrated sensors, may cause problems for the beamformer. In some applications we do have some knowledge about the field, which can be applied to improve the performance of the beamformers. Our program will try to cover all these specialties, and will be used to determine weaknesses and strengths of the different beamformers. The wave field generator is included in the appendix, while all the tested methods are being referred to its original source. The default setup is a uniform linear array (ULA) with ten sensor elements that samples the field perfectly without any aliasing effects. Everywhere where the diagonal loading is used to increase the robustness, is the same small diagonal loading factor used. In this thesis we concentrate on narrowband signals to be able to use the phase difference between the sensors to calculate the delay according to the steering vector. This makes us independent of time samples when we are making delays for each sensor weight. With the knowledge that a broadband signal can be divided into several narrowband signals, we are then able to generalize our results to broadband signals as well.

3.2 Results from the IAA-APES Beamformer

3.2.1 Resolution

As mentioned earlier the IAA-APES beamformer has a better resolution than a DAS beamformer with uniform weights, as illustrated in the figure (3.1). The illustration shows the DAS steered response, and it is not affected

by the field's two sources that are separated with the resolution threshold for the IAA-APES beamformer. We consider a uniform linear array of ten elements; we then achieve a resolution of 3.83° degrees with IAA-APES, compared to 16.9° degrees for the simplest DAS beamformer in presence of coherent sources. This results in a resolution advantage for the IAA-APES with a factor of 4.41 compared to the standard DAS beamformer. Implying that the standard DAS beamformer must have approximately 44 elements in the array to achieve the same resolution. Consequently it will increase the data amount needed to obtain the same resolution as the IAA-APES.

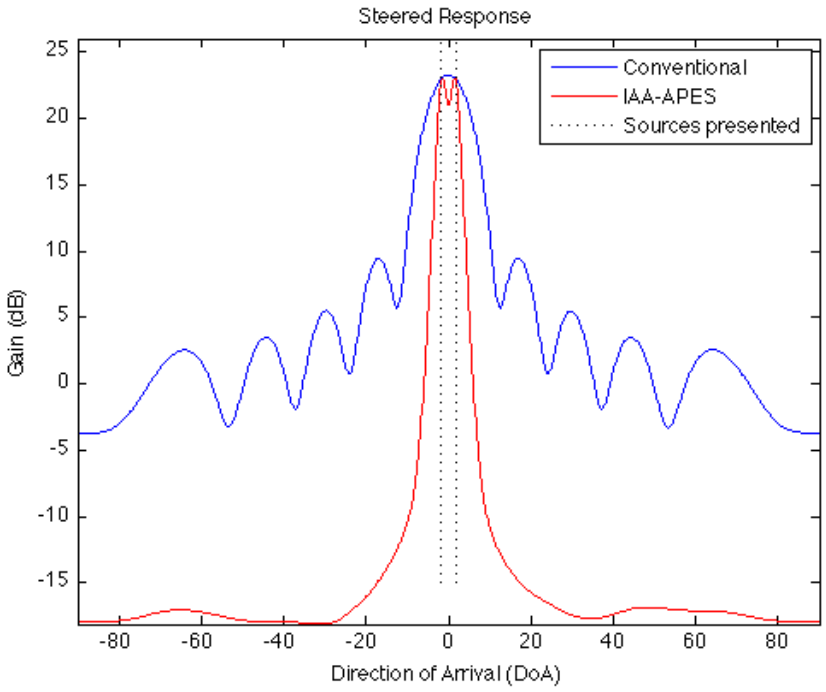


Figure 3.1: IAA-APES resolution compared to a DAS beamformer with uniform weights.

By comparing the resolution of IAA-APES with the Capon beamformer, which is known for having a good resolution, will give an indication of how good the IAA-APES performance is. When we make the comparison with the Capon beamformer we use subarray averaging and diagonal loading to obtain the maximum resolution while still being able to suppress coherent sources without signal cancellation. The choice of subarray length L is a trade-off between resolution or number of degrees of freedom and the ability to decorrelate. A short subarray, e.g. $L = 2$, has low resolution and only one degree of freedom and ability to decorrelate. When using a small subarray the ability to decorrelate sources increase and the signal cancellation effect is reduced, since number of subarray that are averaged increases, $K = M - L + 1$. The full array Capon ,$L = 10$, used on a field of coherent sources, shows the bad ability to decorrelate. The resolution of Capon with full array in presence of coherent sources perform worse than

the DAS beamformer (3.2). The IAA-APES algorithm achieves the same resolution, with or without coherent sources, while the Capon beamformer has two different results of resolution for the two cases of coherence. The Capon with subarray averaging perform better when there is no coherence between the sources. Figure (3.2) illustrates the two cases of coherence, and shows that IAA-APES has a better resolution than the Capon beamformer in presence of coherent sources. When Capon utilize the full array and the sources are incoherent, it has a better resolution performance than IAA-APES. It is common and reasonable to choose the subarray length to be half of the full array $L = M/2$ [16], the choice is because of the tradeoff between spatial extent/resolution and the degrees of freedom. In addition it will give full rank of the estimated covariance matrix.

Overall will IAA-APES perform better than the Capon beamformer with subarray averaging, IAA-APES is able to decorrelate coherent sources without sacrificing the resolution or the number of degrees of freedom, and it deliver a stable resolution performance whether the sources are coherent or incoherent.

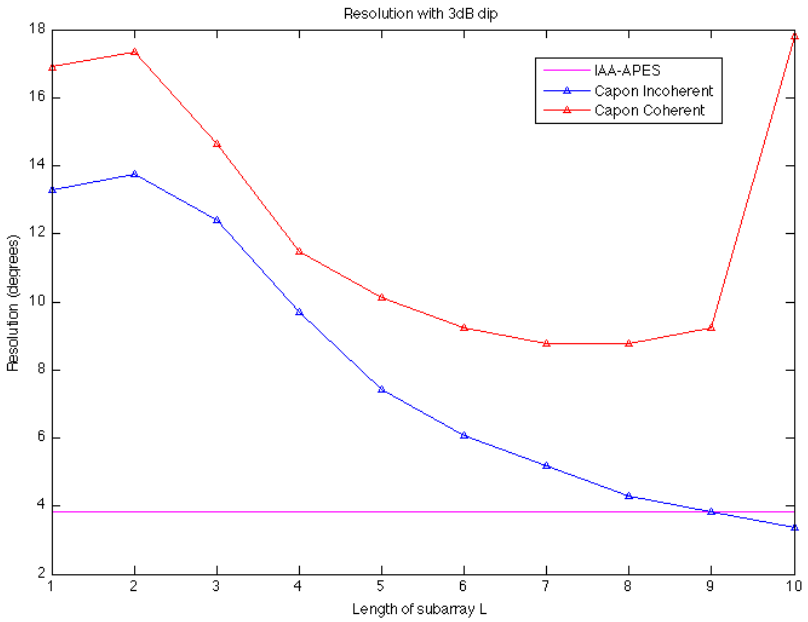


Figure 3.2: IAA-APES resolution compared to a Capon beamformer with subarray averaging.

3.2.2 Error in Direction of Arrival

To determine the direction of arrival we extract the maximum value or the peak in the region of interest. The peak can be misleading and give the wrong estimate of the direction, due to errors in the steering vector $\vec{d}$, or if there are sources close to the signal of interest. In the case of the IAA-APES beamformer, the illustration (3.3) shows that if one interfering source is

close to the desired source, the estimation of direction of arrival gets worse. Interfering sources that is closer to the desired signal than the resolution threshold will never be interpreted as two sources, but will be interpreted as noise that disrupts the estimation of angle of arrival.

The illustration (3.3) shows that the estimation of direction of arrival improves as the interference moves further away from the source of interest. For coherent sources the estimated direction is significantly poorer than for the incoherent case. It is approximately one degree difference between the two different cases of coherence, when the interfering source is close to the desired. When the interference is about twelve degrees away from the desired, the angle deviation becomes 0, and the inference is not affecting the desired mainlobe anymore.

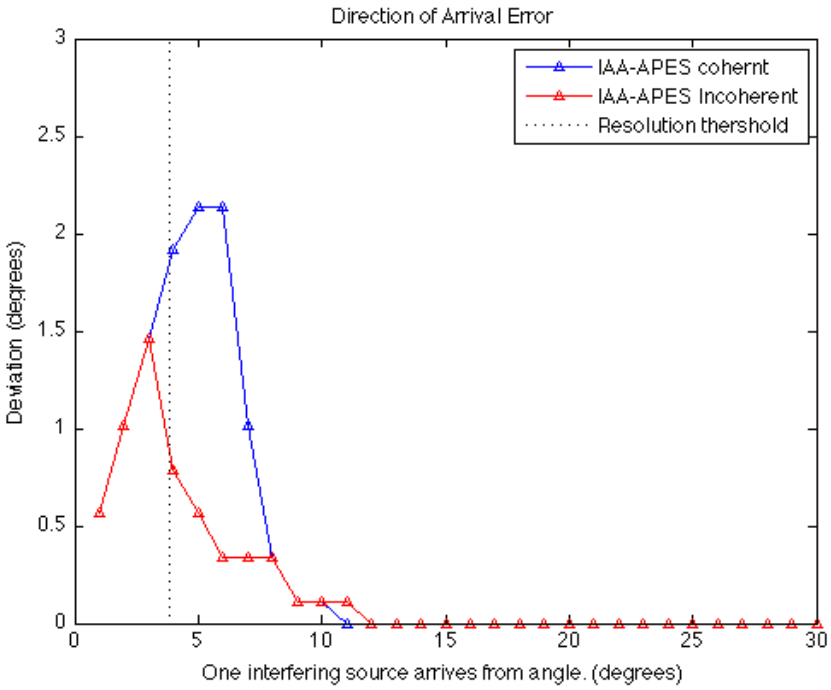


Figure 3.3: Error in direction of arrival presented with one other other source

When the inference is close to the desired source will the direction estimate deviate from the actual direction, the sources will although be resolved since the dip between them is $-3dB$. This effect is illustrated in the figure (3.3), where the desired signal arrives from $0°$ and the interference source $5°$ degree. The interference deform the the desired source's mainlobe, because of leakage from the interference that affect the direction estimate. As seen in the figure (3.4) the peak of the mainlobe is misleading, it is not placed in the direction of the desired source, which causes an inaccurate estimate of direction of arrival. One thing to notice is that if the DOA estimate is wrong, will the estimated covariance matrix $\hat{\mathbf{R}}_{iaa-apes}$ also have the same error because it uses the power estimate as

weights, which can lead to incorrect placement of the zeros.

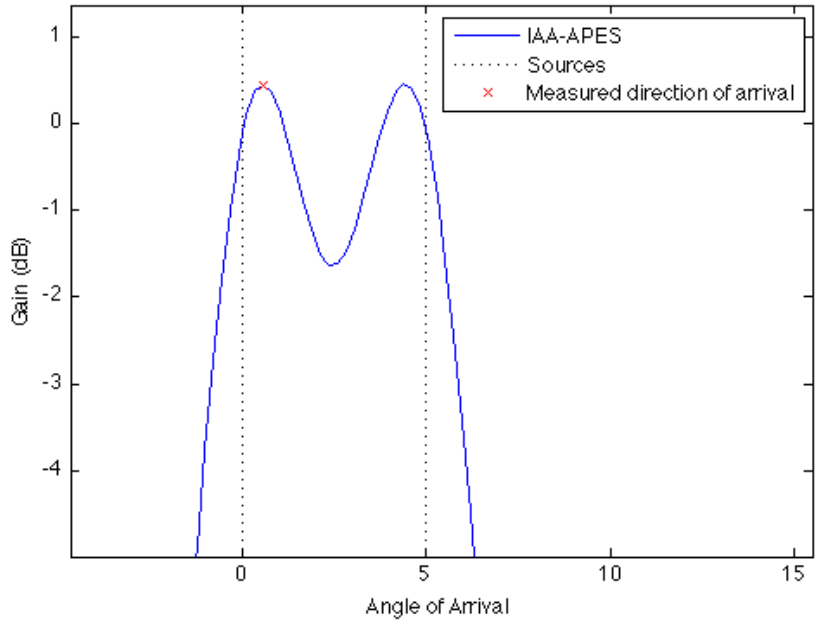


Figure 3.4: Example of estimation error of direction of arrival

3.2.3 Numerical Complexity

By analyzing the algorithm of the IAA-APES beamformer, we will determine how many arithmetic operations that are required for the beamformer design calculations. It gives an indication of how fast the different beamformer methods will operate, and in addition it will give benchmark for how fast it is compared to other methods.

Initial Stage

The initial stage of the IAA-APES method uses a standard DAS beamformer with non-adaptive weights. The DAS beamformer consists of a data matrix with $M \times N$ data inputs and the data matrix is multiplied with a uniform weight vector $\vec{w}^H$. The operation requires $M \times N$ multiplications for every scan-line, $g = 1, \ldots, G$. We assume that the number of time samples, N ,and number of scan-lines, G, are the same for every tested beamformers so they can be excluded in the initial stage. As a result, the numerical complexity of the initial stage using the DAS beamformer with predetermined weights, is proportional to the number of elements in the array, $O(M)$ [3].

Main Iteration

The core of the IAA-APES method is an iterate update of the estimate of the sample covariance matrix $\hat{\mathbf{R}}_{iaa-apes}$ by using the angle matrix $A(\theta) =$

$[\vec{a}(\theta_1), \vec{a}(\theta_2), \ldots \vec{a}(\theta_G)]$, weighted by the power estimates $\hat{P}_g$. Construction of the sample covariance matrix is defined as;

$$\hat{\mathbf{R}}_{iaa-apes} = A(\theta) diag(\hat{P}) A(\theta)^H \tag{3.1}$$

where *diag* function places the power estimates along the main diagonal, resulting in a $G \times G$ matrix. The total number of arithmetic operations performed in the making of the covariance matrix estimate is $M^2 \times G \times I$. The next estimate of $\hat{P}$ can be calculated, and ready for the next iteration. Since the matrix is diagonal constructed with the power estimate as weights, will the resulting matrix has Toeplitz structure, which can be utilized when inverting the estimated covariance matrix $\hat{\mathbf{R}}_{iaa-apes}$. I defines the number of *iterations* it takes to converge the estimates to a satisfactory estimate. $\hat{P}$ requires the inversion of $\hat{\mathbf{R}}_{iaa-apes}$, which again is multiplied by a propagation vector for every angle θ_g.

$$z = \frac{a(\theta_g)^H \mathbf{R_{iaa-apes}}^{-1} \vec{y}[n]}{a(\theta_g)^H \mathbf{R_{iaa-apes}}^{-1} a(\theta_g)} \tag{3.2}$$

Inversion of $\hat{\mathbf{R}}_{iaa-apes}$ is done in $O(M^2)$ arithmetic operations using Levinson- Durbin (LD) recursion. The LD recursion method is utilized due to the advantage it has when applied on a positive definite Toeplitz matrix. It is necessary to do the inversion on every estimate of the covariance matrix $\hat{\mathbf{R}}_{iaa-apes}$, inverting is a time consuming task and has to be done for every iteration. For other adaptive beamformer methods the inversion only has to be done once or twice during the beamforming. The number of iterations I is important for the IAA-APES beamformer, and it has a key value for the numerical complexity. As mentioned have empirical tests shown that after 15 iterations ($I = 15$) [36] the improvement of the performance does not change significantly. The total number of multiplications for making a new power estimate is then $O(M \times N \times G \times I)$.

Total Arithmetics Operations

Algorithms with combined subset of operations, will the the highest order of operations dominate. The dominating highest order will in the case of IAA-APES be to construct the new estimate of the power $\hat{P}$. The number of data samples N will be lager than the number of elements in the array $N > M$, as the rule of thump states that the number of data samples should be $N \geq 2M$ [30]. Because of this we can say that the construction and inversion of $\hat{\mathbf{R}}_{iaa-apes}$ has a lower amount of operations, than constructing the new power estimate $\hat{P}$. As in the initial stage the numerical complexity is proportional to the number of sensor elements M. We can therefore conclude that the resulting numerical complexity for the IAA-APES beamformer is $O(M \times N \times G \times I)$. The IAA-APES iterates to find the weights, every iteration can be seen as a any other adaptive beamformer calculation, which includes inversion of the

Table 3.1: Numerical complexity of IAA-APES

% Initialization by DAS
for all θ **do**
$\hat{P} = \frac{1}{N}\sum_{n=1}^{N} |w_m^H y(n)|^2 \quad O(M)$
end for

% IAA-APES convergence
while convergence **do**
$\mathbf{R_{iaa-apes}} = A(\theta)\hat{P}A(\theta)^H \quad O(M^2 \times G)$
$\mathbf{R_{iaa-apes}}^{-1} = inv(\mathbf{R_{iaa-apes}}) \quad O(M^2)$
for all θ **do**
$z = \frac{a(\theta_g)^H \mathbf{R_{iaa-apes}}^{-1} y(n)}{a(\theta_g)^H \mathbf{R_{iaa-apes}}^{-1} a(\theta_g)} \qquad n = 1, \cdots, N$
$\hat{P} = \frac{1}{N}\sum_{n=1}^{N} |z|^2 \quad O(M \times N)$
end for
end while

estimated covariance matrix required to calculate the weights, and the final calculation of the power estimate.

The IAA-APES method can be programmed so that the different operations can be solved on parallel hardware for instance a GPU (Graphic Processing Unit), this is achievable due to each scanned line θ_g can be computed separately [36]. By taking advantage of this application the computational time can be decreased by distributing the computational tasks, so each scan line can run in parallel.

3.2.4 SNR Related Metrics

All SNR related metrics, *SNR*, *RMSE*, and *array gain* are gather in one section. The results have turned out to that they all three metrics are directly related, because of the distortionless response constraint. Comparing the results with a full array DAS, and Capon using subarray averaging and diagonal loading will give us an indication of the performance. In the illustration (3.5) the choice of subarray length for Capon is put to $L = M/2 = 5$. It seems that IAA-APES and DAS have a better array gain (3.5a) for low SNR values than Capon (subbarr.avg L=M/2), this is an effect caused by the number of elements in the array, where IAA-APES and DAS uses a full array, while Capon only uses $L = M/2$. Assuming spatially white noise, the SNR related metrics are all proportional to the array length. Immediately after the signal power is stronger then the noise power (SNR > 1) the Capon beamformer deviates from the DAS beamformer. Without the array length disadvantage the Capon beamformer will converge towards the DAS beamformers for low SNR values, this was proved in (1.42). The performance of the SNR related metrics is basically depending on two parameters; coherence and the separation between the desired and the interfering signal. We will cover both cases to give a good impression and comprehension of the SNR related performance for the IAA-APES method. It will

attain a better performance than DAS in all SNR related metrics, but does not carry out the great performance to Capon with subarrays, in presence of incoherent sources that are properly resolved.

Since the sampled field is random it is natural to test for coherent sources. The result shows that IAA-APES has a strong and significant performance in presence of coherence in the wave field (3.6). It attains the Capon performance and even exceeds Capon for high SNR values.

When the Capon method is applied on coherent sources it does not reach the same gain as in the incoherent case, this indicates that the Capon is somewhat weaker with coherence. The IAA-APES on the other hand does perform better than Capon in case of coherence and SNR values higher than 40 dB. We have now presented the two cases, incoherent and coherent sources that are properly resolved. Testing how the performance will change when we are decreasing the spacing between the desired and the interfering source is the next thing we need to study in more detail. We set a fixed SNR value at 30 dB and then vary the direction of arrival of the interfering source. As mentioned the IAA-APES beamformer performs better, and is more stable and robust than the two other beamformers, Capon and DAS. We examine the case of coherence, and it shows that the IAA-APES method has a better and more stable estimate of the amplitude (3.7b) as expected. The variations in gain is significantly lower for IAA-APES than for the other beamformers (3.7), and it is also more independent whether the source are coherent or not, which makes IAA-APES a more stable beamformer.

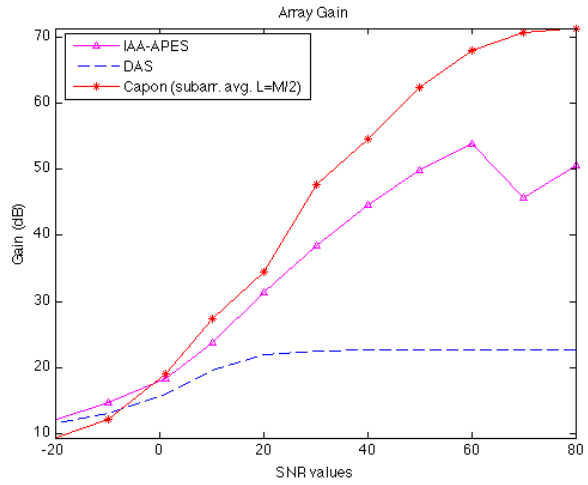
The difference between the incoherent and coherent cases for a fixed value of SNR together with changing direction for the interfering source, is that Capon performs poorer in the case of coherent sources. This difference is illustrated (3.8) with the SNR metric for both cases of coherence.

3.2.5 Coherent versus Incoherent Signals

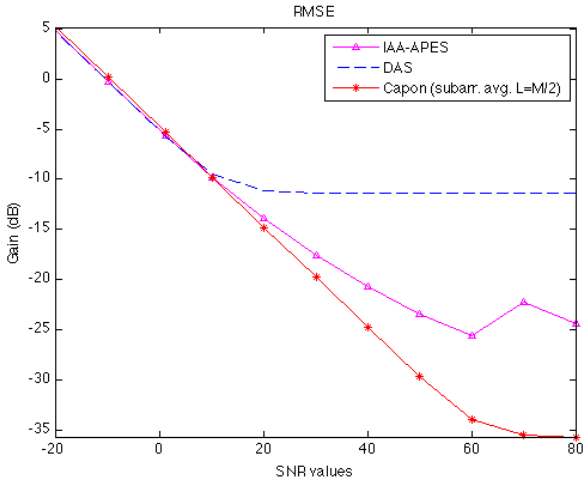
The IAA-APES method manages the two cases of signal coherence well. The resolution is the same in both cases, and the SNR related metrics are not remarkably affected. The initial stage of IAA - APES in presence of coherence starts a bit further away from the convergence point, this is triggered by coherence and closely spaced interfering sources, illustrated in (3.9). Although the initial power estimate $\hat{P}$ is further away from the convergence threshold, will the coherent case use approximately the same number of iterations to reach the convergence threshold as the incoherent case.

3.2.6 Number of Time Samples

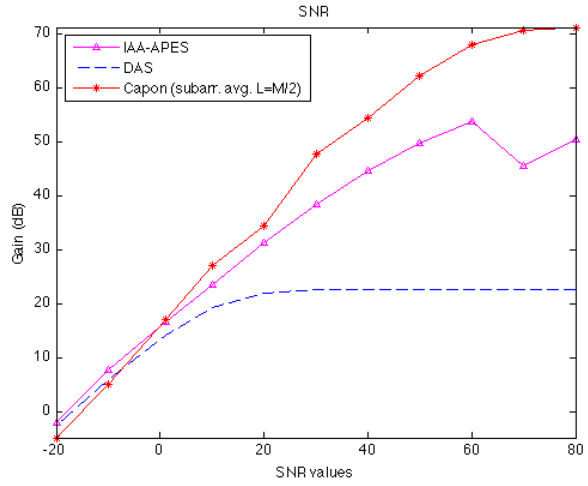
For Capon beamformer we usually have to estimate the second-order statistics from a finite amount of data, which makes the estimates of covariance matrix dependent on the number of data samples N [30]. This is not the case for the IAA-APES where the construction of the covariance matrix is done iteratively with the signal power as weights. The initial



(a) Array gain

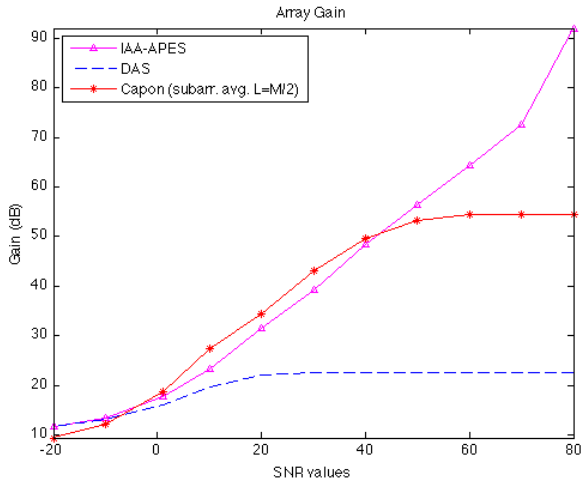


(b) RMSE

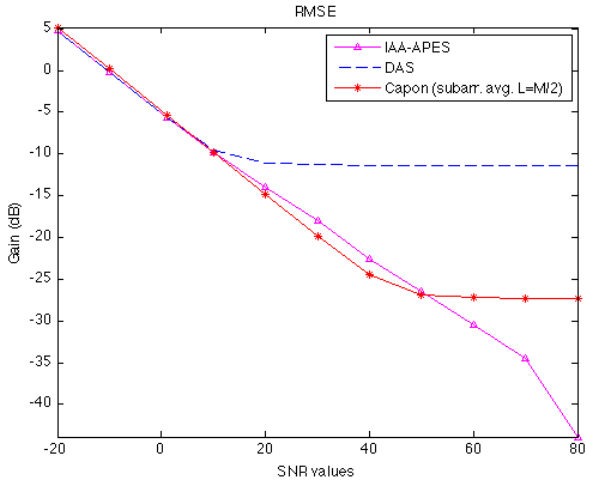


(c) SNR

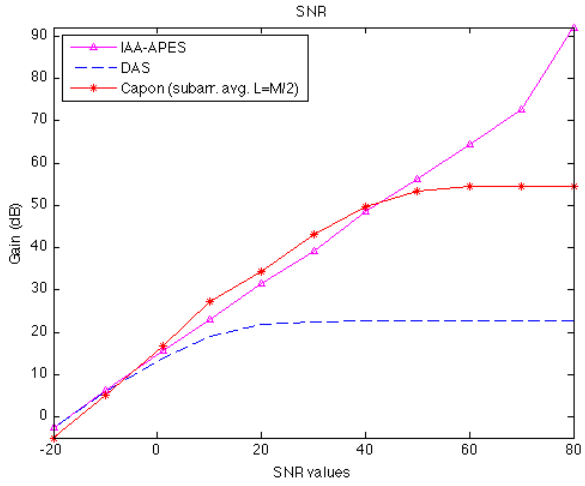
Figure 3.5: SNR related metrics in presence of *incoherence* and properly separated sources (40 degrees).



(a) Array gain

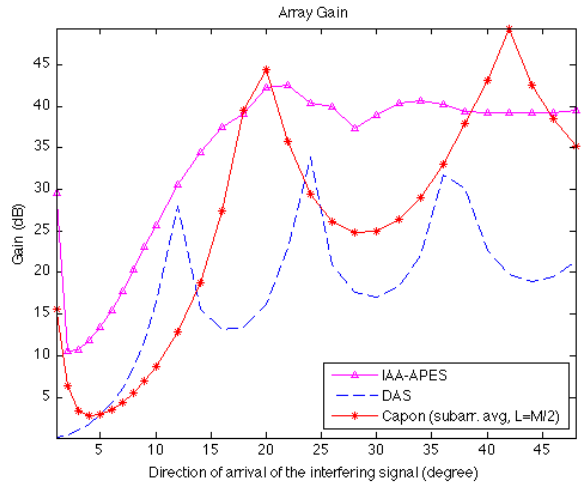


(b) RMSE

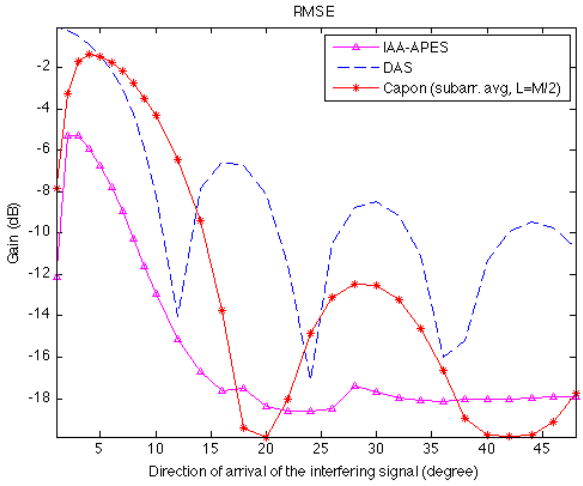


(c) SNR

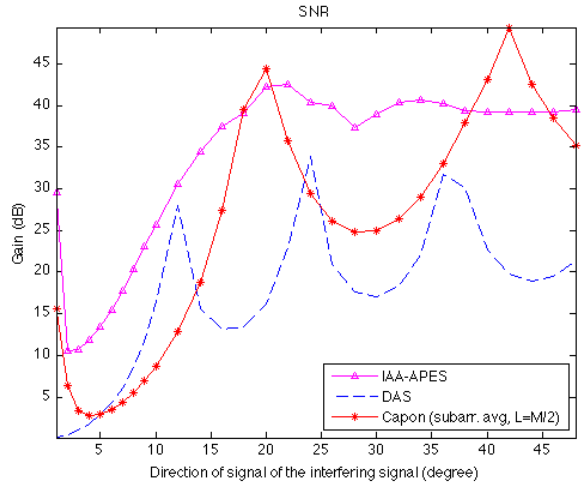
Figure 3.6: SNR related metrics in presence of *coherence* and properly separated sources (40 degrees).



(a) Array gain

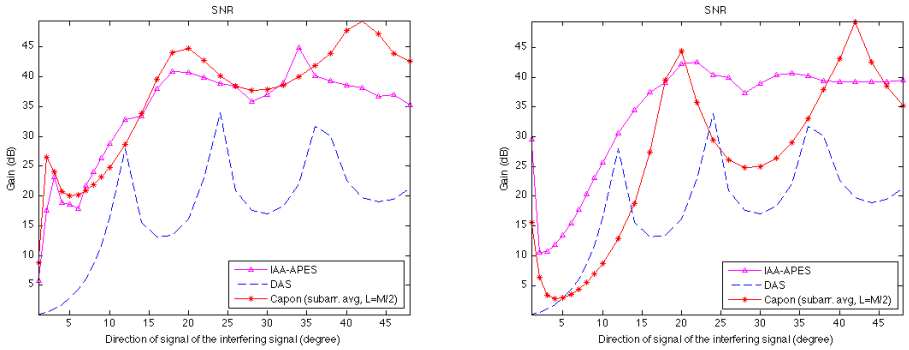


(b) RMSE



(c) SNR

Figure 3.7: SNR related metrics in presence of *coherence* between the sources, fixed SNR value 30 dB, and vary the direction of arrival.



(a) SNR incoherent sources (b) SNR coherent sources

Figure 3.8: Comparison between incoherence and coherence for fixed SNR and one interfering source

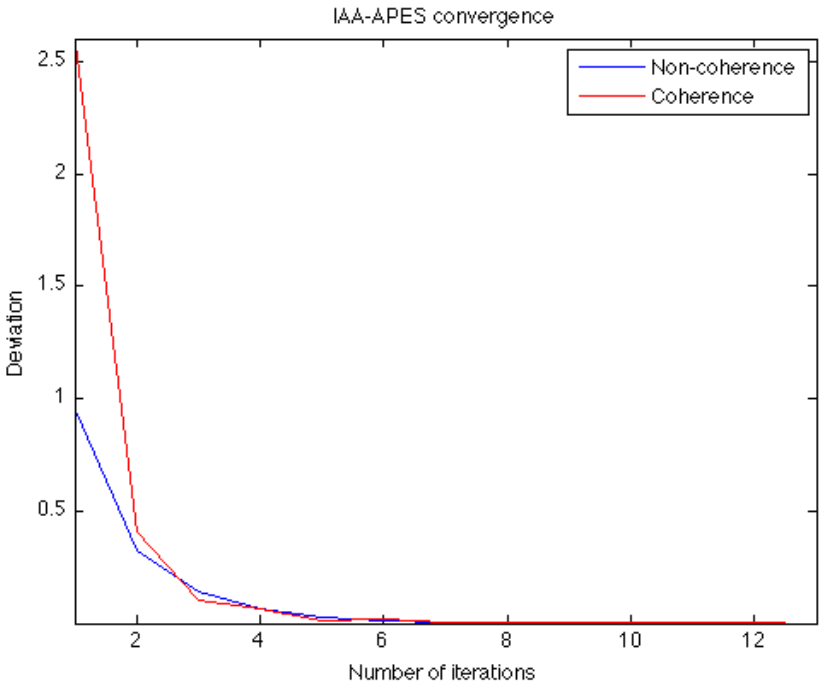


Figure 3.9: Convergence comparison of both of cases coherence

stages of IAA-APES is depended on the number of samples N, to construct the initial power estimate $\hat{P}$. Is the rest of the IAA-APES method dependent of the number of time samples N? We know that IAA-APES is dependent on angle resolution, or the number of scan lines, to be able to construct the estimated covariance matrix. To investigate the dependency of the number of time samples N a plot of the SNR value is made. On x-axis is the time samples N, and on the y-axis is the direction of the interfering source. As illustrated (3.10) the SNR performance increase in the interval of [N=1,2,...,20]. For $N > 20$ the performance does not improve significantly,

and will consume meaningless data power without benefits. The blue area in the middle of the image indicates the resolution of IAA-APES.

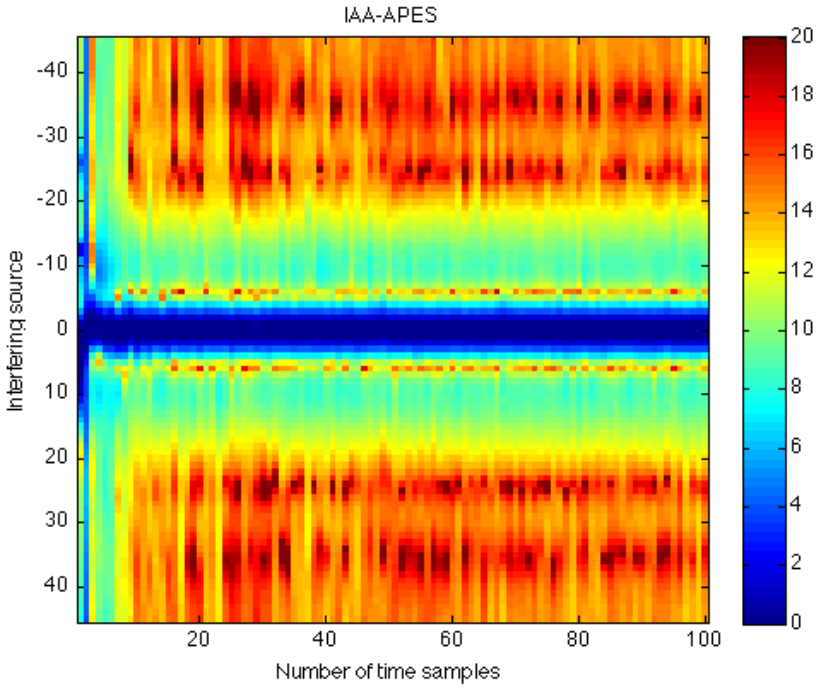
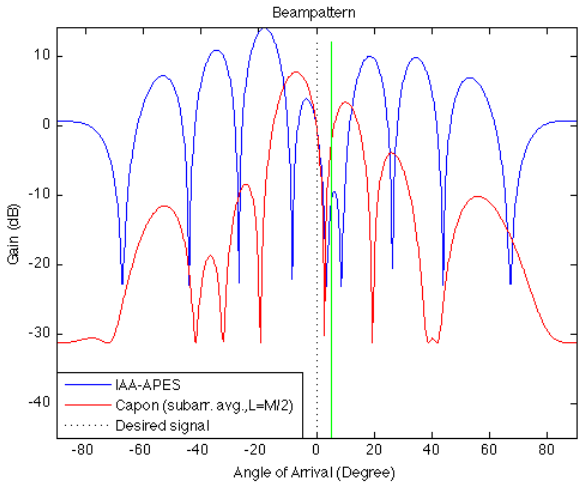


Figure 3.10: Using the SNR metric to show IAA-APES dependency of time sample N.The SNR values are showed in dB scale.

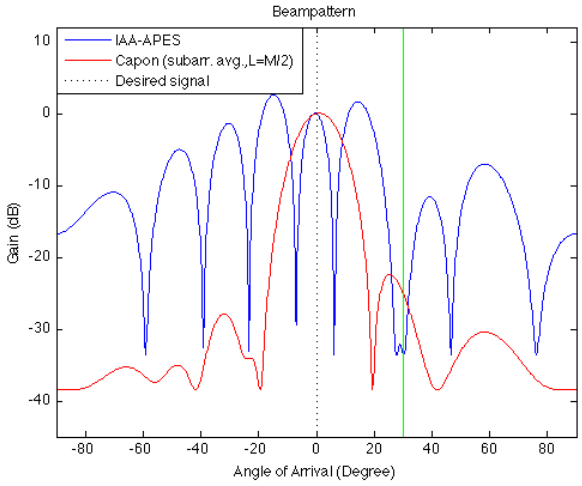
3.2.7 Attenuation Properties

All beamformers task is to preserve the desired signal and attenuate other interfering signals. We investigate how the beampattern will handle different scenarios. Our simulated field consist of signals that are coherent with the desired signal, signals that are coherent but not with the desired, and signals that are incoherent. By varying the number of sources, and theirs angle of arrival can give an impression of of the attenuation properties. Comparing the attenuation performance with the Capon with subarray length $L = 5$, since $L = 5$, it will have only 4 degrees of freedom to suppress unwanted signals. Whereas IAA-APES uses a full array, and then have more freedom degrees to place zeros to suppress unwanted signal. The term degrees of freedom means the number of weights that can be used to suppress interference. When only one interference is present in the field the IAA-APES has a narrow mainlobe compared to Capon with subarray averaging. The IAA-APES allows the sidelobe increase above the response in the steering direction 0°, since it knows that there is low energy, or no sources in that sidelobe area. The figure (3.11) shows two illustrations; one interference approaches the mainlobe and one interference in the sidelobes.

When the interference approaches the main lobe, there are mainly two effects;



(a) Interference approaching the mainlobe of IAA-APES



(b) Interference in the sidelobe region

Figure 3.11: Beampattern of IAA-APES with one interfering source compared with Capon with subarray averaging.

1. The mainlobe is shifted away from the steering direction, and its heights is larger than unity.

2. The height of the sidelobes closest to the steering direction increases.

The beampattern allows the mainlobe peak to be shifted away from the steering direction to another direction where there is no signal, and rather concentrate to fulfill the distortionless response criteria, $\vec{w}^H \vec{d} = 1$.

When the interference is inside of the mainlobe, the beamformer splits the mainlobe into two "sidelobes" where the heights are higher than the signal direction heights. The peak of the pattern is no longer in the steering direction.

The figure (3.12) shows that the IAA-APES is able to suppress coherent sources (green) as well as incoherent (black). The magenta colored sources

are coherent with each other but not with the signal of interest in $0°$. The figure shows that both beamformers have preserved the distortionless response, by keeping unit gain in the steering direction $\vec{w}^h\vec{d} = 1, (db(1) = 0)$. The Capon with sub-averaging suppresses the interference more than IAA-APES, but both beamformers put the zeros in close to the direction of the interference.

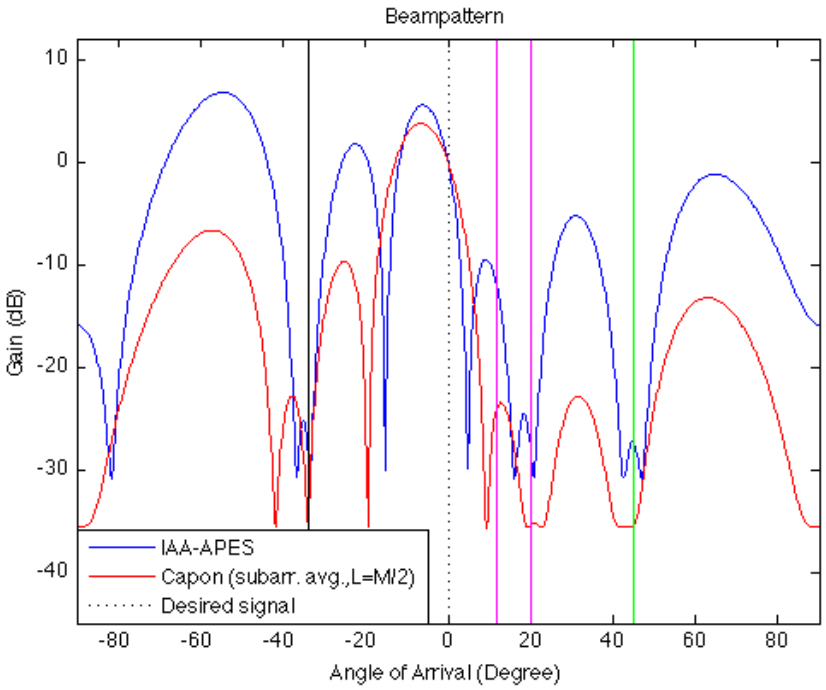


Figure 3.12: The beampattern of a IAA-APES in presence of five sources in the field.

3.2.8 IAA-APES Summary

The IAA-APES method has an overall suitable performance, and performs well in both cases of coherence, which gives the IAA-APES an advantage on Capon with subarray averaging. The resolution is the same for both cases of coherence, and the SNR related metrics do not vary as much as Capon with subarray averaging. An advantage of using IAA-APES is its high resolution performance, together with a largest number of degrees of freedom that is available. The IAA-APES is an iterative method, which requires repeated calculation for the adaptive weights, which makes the numerical complexity high. We can make the conclusion that IAA-APES is a stable and robust method, with a large number of degree of freedom to suppress unwanted signals, but has high numerical complexity.

3.3 Results of Adaptive Spatial Averaging

3.3.1 Resolution

The adaptive spatial subarray averaging provides a resolution that is better then the DAS beamformer, it is derived from the standard subarray averaging, but makes the structure of covariance matrix $\hat{\mathbf{R}}$ close to Toeplitz by applying adaptive weights. How does the adaptive weights that are applied to the estimated covariance matrix affect the resolution? The adaptive spatial averaging provides a high resolution, and has improved resolution compared the resolution performance of the Capon with standard subarray averaging. Another competitive advantage of adaptive spatial averaging is that the method is not affected by coherence, such as standard subarray average that operates with two results of resolution. The illustration (3.13) shows results of different lengths of subarray, L, and it shows that the adaptive spatial averaging obtain a better resolution the Capon with subarray averaging. The adaptive spatial averaging method is constrained so that the measured peaks are approximately equal in gain. This was done to avoid one peak dominating the other completely. This may be the reason why the adaptive spatial act strange for subarray length $L = 6, 7, 8, 9$. Without the peak constraint the adaptive spatial averaging has the same structure of the curve as the incoherent case of standard subarray averaging. The adaptive spatial averaging uses a two-layer optimization, that makes this method more sensitive compared to other single optimization methods.

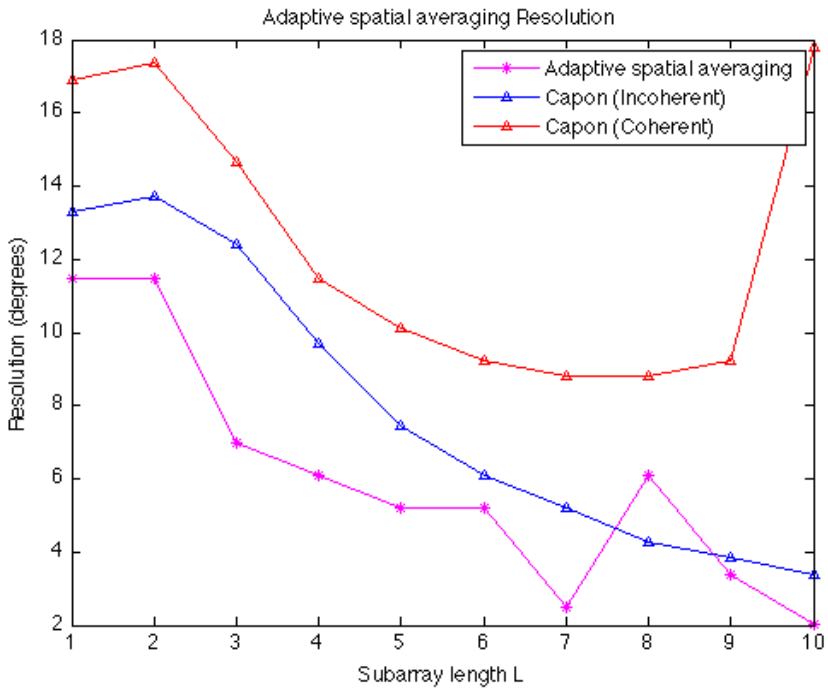


Figure 3.13: Resolution of the adaptive spatial averaging

Determination of subarray length is a compromise between achieving high resolution or higher degrees of freedom to suppress unwanted sources and ability to decorrelate, as mention earlier.

3.3.2 Error in Direction of Arrival

Investigating the accuracy of the measured direction of arrival can decide how much we can rely on the estimated direction of arrival. The deviation between the measured and the actual direction of arrival is desirable to keep low as possible. To measure the direction of arrival error an interference is varied from 1° to 45°, and then calculate the deviation between the measured and the actual direction. Figure (3.14) shows the interference close to the desired source in 0°, it will then affects the mainlobe resulting in a larger error in the estimate. The beamformer will not perceive two source until the interference has passed the resolution threshold, only disrupt the direction estimate. The coherent interference will have a higher deviation after exceeding the resolution threshold than the incoherent interference, which may lead to incorrect estimates of the direction of arrival. When the interference angle exceeds 15°, the deviation is zero for both cases of coherence, and the interference is not manipulating the desired mainlobe.

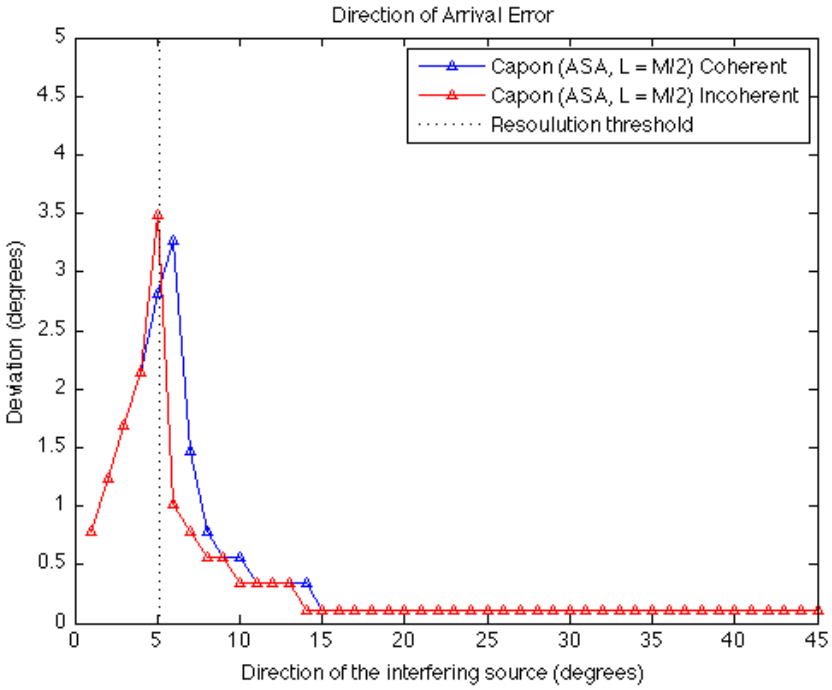


Figure 3.14: Error in direction of arrival for the adaptive spatial averaging

3.3.3 Numerical Complexity

The numerical complexity of the adaptive spatial averaging has similarities to the original subarray averaging. Every k sample covariance matrix is first constructed so that we can find $\hat{\mathbf{R}}_{sa \cdot k}$. The number of arithmetic operations to construct these matrices is $O(N \times L^2 + K - 1)$, due to the overlapping subarrays. The term $K - 1$ is so small that it can be ignored, and we get $O(N \times L^2)$ arithmetic operations for constructing $\hat{\mathbf{R}}_{sa \cdot k}$. To construct the adaptive weights for the estimated covariance matrix $\hat{\mathbf{R}}_{sa \cdot k}$, is it the deviation matrix required, where each element in the matrix describes how much that the element deviates from the mean value of its diagonal. Looping through the matrix to acquire the deviation matrix, requires $O(K \times L^2)$ operations. The adaptive weights v_k that are applied to $\mathbf{R}_{sa \cdot k}$ is found by constructing the error matrix $\hat{\mathbf{R}}_{ee}$ that again will need $O(K \times L^2)$ operations. The inverse of $\hat{\mathbf{R}}_{ee}$ is need to calculate the adaptive weights, the matrix has a dimension of $K \times K$, and does not have an optimal structure for inversion resulting in $O(K^3)$ operations. Having the inverse of $\hat{\mathbf{R}}_{ee}$ the weights can be calculated straight forward with $O(2K^2 + K + 1)$, ignoring the small quantities we end up with only $O(K^2)$ operations. The remaining operations involve applying the weights to the subarray averaged matrix $\hat{\mathbf{R}}_{sa \cdot k}$. The calculation is then dependent on how large the subarray length L is, which again determines the size of the matrix $\hat{\mathbf{R}}_{sa}$. The number of multiplications required to multiply a scalar with a matrix is $O(L^2)$. The new adaptive estimated covariance matrix is constructed, the matrix is close to being a Toeplitz matrix, but will still require $O(L^3)$ operations for inversion. The adaptive spatial averaging method involves several calculations compared to the standard subarray averaging, as mentioned earlier the choice of using adaptive weights increases operative time, it is often a trade-off between time and performance. The most time consuming task is $O(L^3)$ or $O(K^3)$ depending on the size of the subarray. The adaptive spatial averaging demands two types of adaptive weights, one for finding the weights v_k that are applied the estimated covariance matrix $\hat{\mathbf{R}}_{asa}$, and one to find the adaptive weights $\vec{w}_m$ that shall be applied to the array output.

3.3.4 SNR Related Metrics

The adaptive spatial averaging provides good performance on the SNR related metrics, SNR,RMSE and array gain. The figure (3.16) shows that the adaptive spatial averaging provides a higher and more stable performance in all SNR related metrics than the Capon with subarray averaging. The direction of arrival of the interfering source does not influence the SNR related metrics, as it does for the standard subarray averaging. The adaptive spatial averaging have the same structure on the curves for both cases of coherence, while the subarray averaging performance varies more in case of coherence between the sources.

It reach the same level of performance for both cases of coherent and incoherent sources.The only different is that the coherent sources initially

Table 3.2: Numerical complexity of Adaptive Spatial Averaging

%% Make covariance matrix
M % Number of elements
L % Subarray length
K = M-L+1 % Number of subarrays
for all K **do**
$\quad \hat{\mathbf{R}}_{sa \cdot k} = \vec{y}_k * \vec{y}_k^H \quad \% \, O(N \times L^2)$
end for

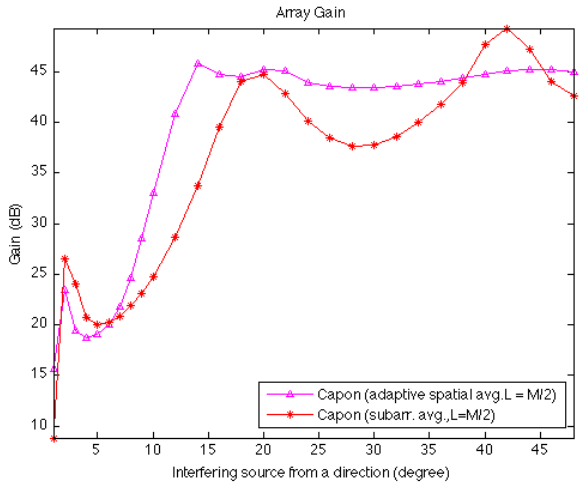
% Make the deviation matrix
$\epsilon = \sum_{i=0}^{L-2} \sum_{l=1}^{L-i} |\hat{r}_{l+i,k} - \hat{r}(i)|^2 \quad \% \, O(K \times L^2)$

% Find adaptiv weights
$\hat{\mathbf{R}}_{ee} = \sum_{i=0}^{L-2} \sum_{l=1}^{L-i} Re\{e_{l+i,l} \cdot e_{l+i,l}^H\} \quad \% \, O(K \times L^2)$
$\hat{\mathbf{R}}_{ee}^{-1} = inv(\hat{\mathbf{R}}_{ee}) \quad \% \, O(K^3)$

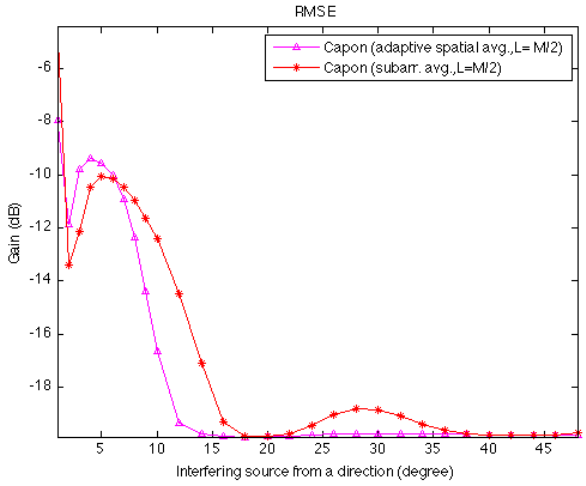
% Calculate weights
$\vec{v}_k = \frac{\hat{\mathbf{R}}_{ee}^{-1}\mathbf{1}}{\mathbf{1}^T \hat{\mathbf{R}}_{ee}^{-1}\mathbf{1}} \quad \% \, O(K^2)$

% Apply the weights on the subarray averaged matrix
$\hat{\mathbf{R}}_{asa} = \vec{v}_k \hat{\mathbf{R}}_{sa \cdot k} \quad \% \, O(K \times L)$

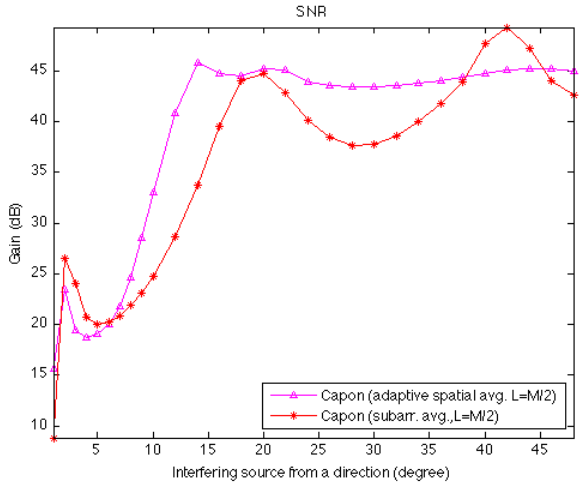
% Inverting $\hat{\mathbf{R}}_{sc}$
$\hat{\mathbf{R}}_{asa}^{-1} = \text{inv}(\hat{\mathbf{R}}_{asa}) \quad \% \, O(L^3)$



(a) Array Gain



(b) RMSE



(c) SNR

Figure 3.15: SNR related metrics for fixed SNR value (30 dB) and different direction of arrival of a incoherent interfering source.

has a better gain when the interfering source is close to the desired source.

3.3.5 Number of Time Samples

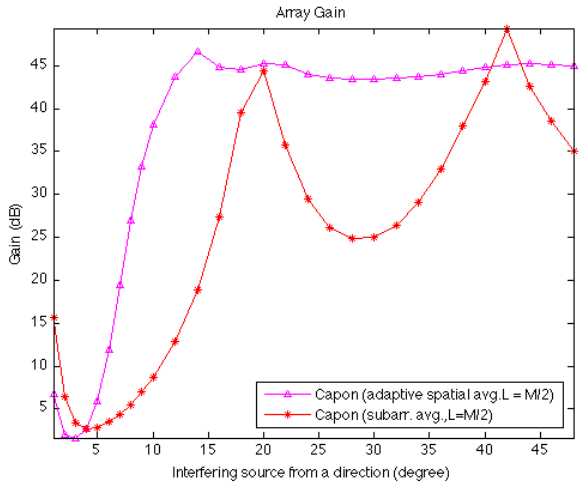
Often the number of time samples ,N, is critical to make a reliable estimate of the covariance matrix, because of the credibility to the statistics increases with the number of time samples. On the other hand we want to figure out which beamformer that has a satisfactory performance with a low number of time samples, due to real time applications and non-stationarity in the observed field. To investigate the performance depending on N, the SNR metric is chosen. While increasing the number of time samples N and changing the direction of arrival of the interference, the performance can be studied. The figure (3.17) shows that the adaptive spatial averaging needs more time samples than Capon with subarray averaging to stabilize its performance. The SNR performance of both cases stabilizes its performance before $N = 20$, and other remarkable notice is that adaptive spatial averaging has a more unstable performance when the source is close to the desired source. The blue area in the middle of both figures indicates the resolution, with the desired signal in the middle at 0° degree.

3.3.6 Attenuation Properties

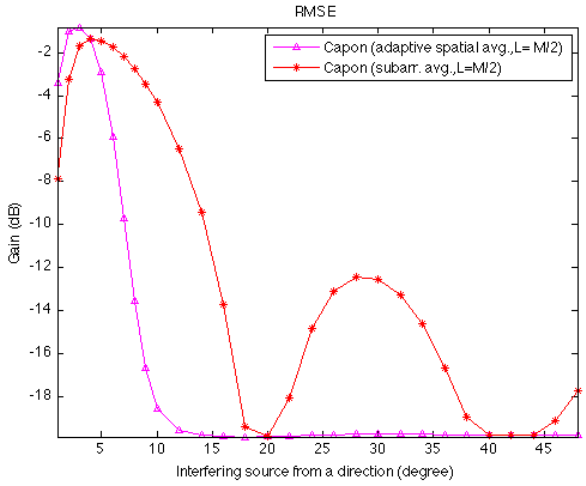
The adaptive spatial averaging has a similar beampattern as the the Capon with subarray averaging, but the adaptive spatial averaging method has a lower gain overall. Although, the adaptive spatial averaging allows greater control to the side of the mainlobe, while the distortionless response constraint is preserved. The figure (3.18) shows the similarities, and shows that the adaptive spatial averaging method is able to suppress coherent sources (green) as well as incoherent (black). The two sources marked with magenta are coherent with each other, but not with the desired source in 0°. Both compared beamformers perform equally on the placement of the zeros in the direction of the interference.

3.3.7 Adaptive Spatial Averaging Summary

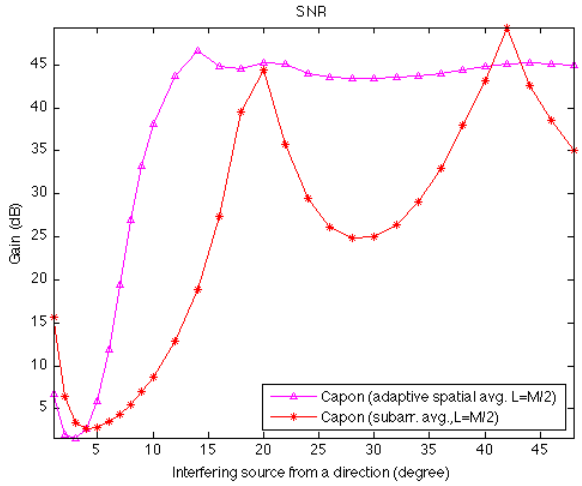
The adaptive spatial averaging has high resolution combined with good performance in the SNR related metrics. The numerically complexity has a higher number of operations than Capon with subarray averaging, because of the adaptive weights that are applied to the estimate of the covariance matrix. The double adaptivity for constructing both weights, increases the numerically complexity, the matrix is not also being totally Toeplitz, so the inversion of the matrix is $O(L^3)$. However, the adaptive weights have a positive impact on the performance that exceeds the performance of standard subarray averaging.



(a) Array Gain

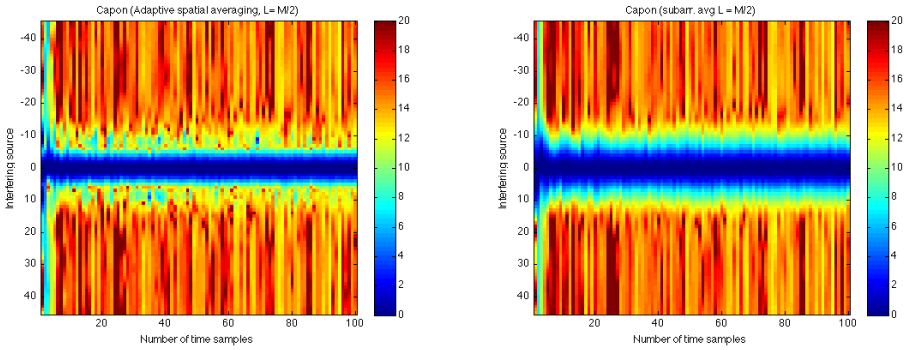


(b) RMSE



(c) SNR

Figure 3.16: SNR related metrics for fixed SNR value (30 dB) and different direction of arrival of a coherent interfering source.



(a) Capon (adaptive spatial averaging, L= M/2)

(b) Capon (subarray averaging, L=M/2)

Figure 3.17: Measuring the SNR for different time-samples and different angle of arrival for one interfering source

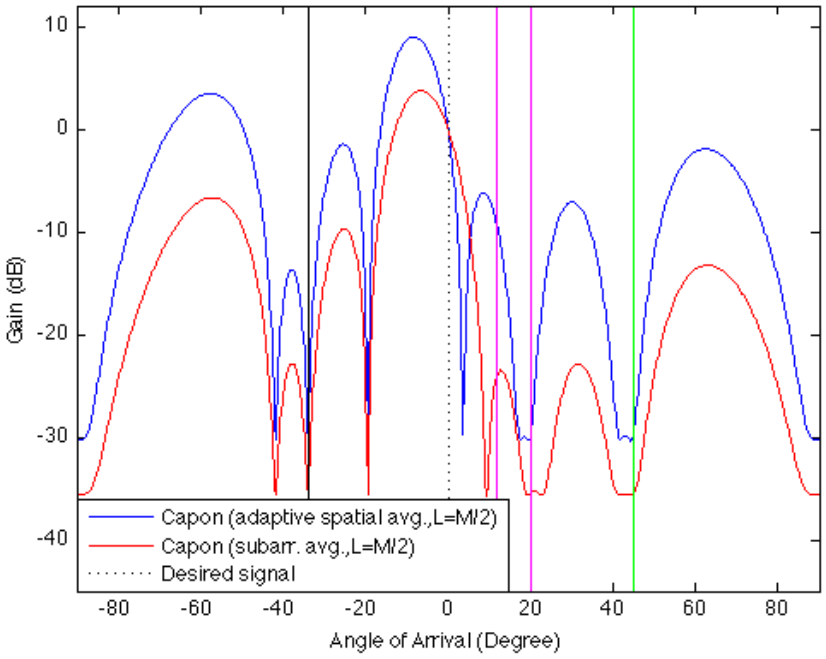


Figure 3.18: Adaptive spatial averaging beampattern compared with the standard Capon subarray averaging.

3.4 Results of the Spatial Convolution

3.4.1 Resolution

When using spatial convolution to make an estimate of the covariance matrix $\hat{\mathbf{R}}_{sc}$, the estimate provides better resolution than a DAS beamformer. But as mentioned, the Capon full array beamformer outperforms spatial convolution method when it comes to resolution. The figure (3.19) The

spatial convolution method results in two different resolutions in presence of both cases of coherence, just as the Capon method (3.19). Comparing the resolution of the spatial convolution method with the resolution performance of the Capon method with a common subarray length of $L = M/2 = 5$, the result shows that the spatial convolution method has a marginally advantage in the case of coherent sources. Remember that the Capon beamformer have reduced the number of degrees of freedom down to four when using a subarray length of L=5. However, the spatial convolution method, on the other hand has not reduced the degree of freedom and have all remaining nine degrees of freedom.

When we have incoherent sources a Capon beamformer with a subarray length of five, will Capon achieve a better resolution than with the spatial convolution method. The difference between the two beamformers resolution performance is greater when the sources are incoherent. The spatial convolution method has a resolution advantage of 0.14° degrees compared to the Capon with a length $L = M/2$ when the sources are coherent. While when the sources are incoherent the Capon method gain an advantaged on 1.94° degrees.

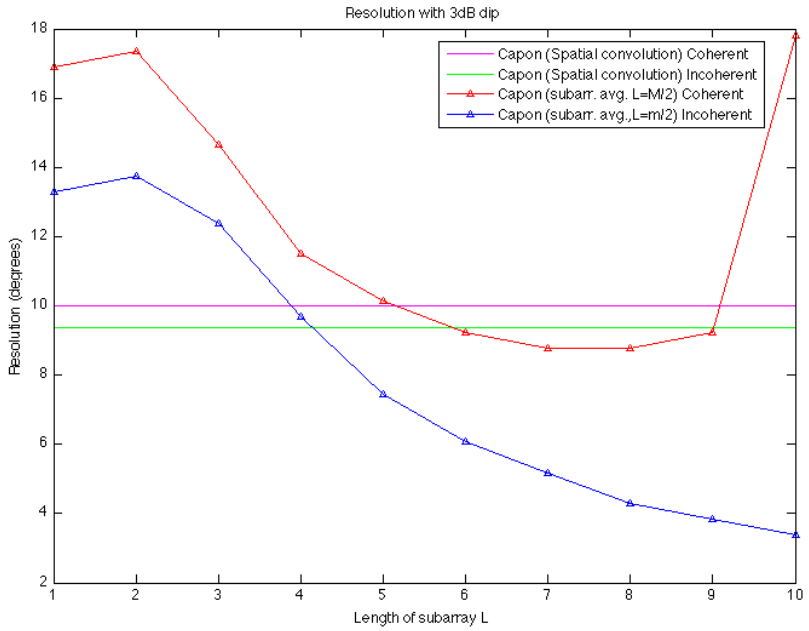


Figure 3.19: The resolution of spatial convolution and subarray averaging with Capon weights

3.4.2 Error in Direction of Arrival

To illustrate the error of arrival we measured the difference between the estimated and the actual direction of arrival. The figure (3.20) illustrate how the bias in the direction change when a interfering source moves outward from the desired source located in 0° degree. The dashed vertical lines

indicate the resolution thresholds, the coherent scenario are marked with blue and the incoherent scenario is marked red. The interfering sources will not be separated if it is located before the resolution threshold, it will only disrupt the desired mainlobe. The figure (3.20) illustrate both cases of coherence, and shows that the spatial convolution method has a larger bias when working with coherent sources. In addition the coherent case have greater variation in the curves, while the incoherent case have smaller variations. When the interfering source moves outward from the desired source the variations in the angle bias tends to degrade. But still the case of coherent sources has a bias in the angle estimate, which can be up to 1° degree wrong. In the article [11] it is shown that redundancy averaging, which has the same effect on the estimate of the covariance matrix as spatial convolution, will lead to a biased DOA estimate. This effect can explain the poor performance of the DOA.

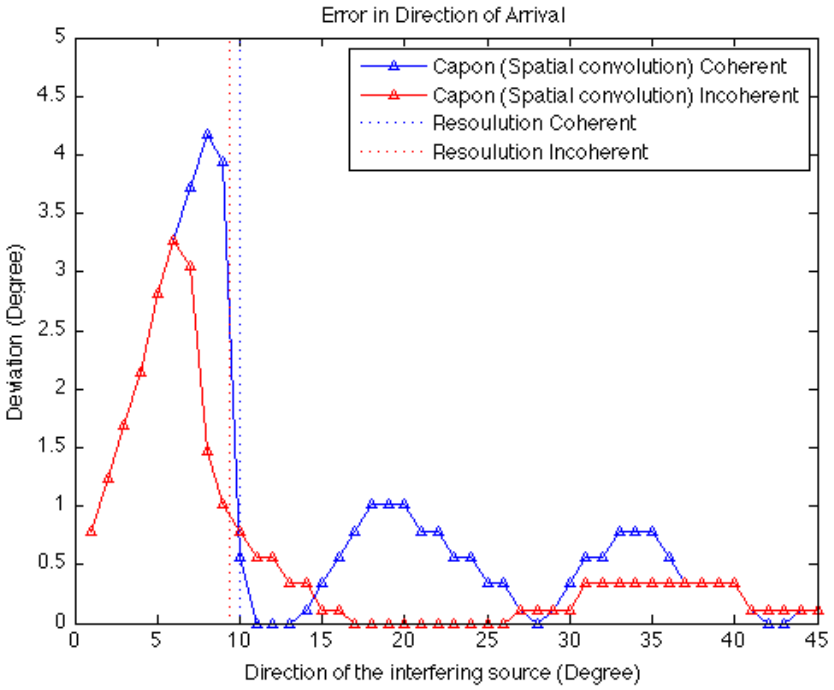


Figure 3.20: Error in the estimated direction of arrival for the spatial convolution method

3.4.3 Numerical Complexity

The numerical complexity of the spatial convolution method can be compared to the standard Capon method with subarray averaging. The difference between the two methods is the zero-padded vector $\hat{\vec{y}}$ is used to make the estimate of the covariance matrix, instead of the original data vector $\vec{y}[n]$. As a result we have an increase in vector size from the original size M to $M_{zp} = 3M - 2$. We use the same procedure as the standard subarray averaging when we are making the estimate of

Table 3.3: Numerical complexity of Spatial Convolution

%% Make covariance matrix
M % Number of elements
L % Subarray length
$K = M_{zp} - L + 1$ % Number of subarrays

% Constructing the zero-padded vector.
$\hat{\vec{y}}[n] = [0, 0, \ldots, 0, y_0[n], y_1[n], \ldots, y_{M-1}[n], 0, \ldots, 0, 0]^T$
for all K **do**
$\quad \hat{\mathbf{R}}_{sc} = \hat{\mathbf{R}}_{sc} + \hat{\vec{y}}_k * \hat{\vec{y}}_k^H / K \quad \% \; O(N \times L)$
end for

% Inverting $\hat{\mathbf{R}}_{sc}$
$\hat{\mathbf{R}}_{sc}^{-1} = \text{inv}(\hat{\mathbf{R}}_{sc}) \quad \% \; O(L^2)$

covariance matrix, but with the zero-padded vector. Applying the zero-padded vector with the new size $M_{zp} = 3M - 2$, to the subarray averaging method with a subarray length, L, equal to the number of sensor elements, M ($L = M$). The resulting number of subarray that is averaged is $K = M_{zp} - L + 1$ which in our general case with ten elements in an array will give $K = 19$. When the subarrays are overlapping it reduce the number of multiplications that is required. The first subarray has L multiplications, and then the remaining $K - 1$ subarrays have only one new multiplication for each subarray, because of the overlapping subarrays. The total number of arithmetic operations for constructing the covariance matrix $\hat{\mathbf{R}}_{sc}$ with the spatial convolution method is expressed as $O(N \times L + K - 1)$. The last term is negligible and can be removed, the total operations for constructing the $\hat{\mathbf{R}}_{sc}$, is $O(N \times L)$ when utilizing the overlapping subarray.

To calculate the weights, the inverse of the estimated covariance matrix is needed,$\hat{\mathbf{R}}_{sc}^{-1}$. The spatial convolution method makes the covariance matrix Toeplitz, which will be utilized so the inversion can be done with $O(L^2)$ operations, by the Levinson-Durbin algorithm. The method is chosen because of the Toeplitz structure of the covariance matrix. The standard subarray averaging needs $O(L^3)$ operation to determine its inverse. The subarray length L determines the dimension of the resulting matrix. In the case of the spatial convolution, L is equal to the number of elements M in the array. The remaining rest of the beamformer is similar to a standard Capon beamformer. For that reason, the main part of interest is the construction and calculation of the weights, where the most heaviest operation is the inversion of the covariance matrix $\hat{\mathbf{R}}_{sc}$, which describe the numerically complexity of the the spatial convolution, with its $O(L^2)$ operations.

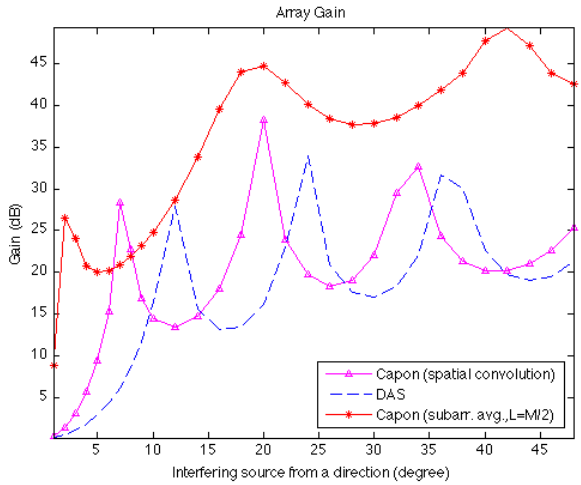
3.4.4 SNR Related Metrics

An investigation of the spatial convolution method with respect to SNR related metrics, will give a deeper understanding of the beamformers performance. The metrics we have focused on is SNR, RMSE and array gain. First we will illustrate the performance by setting a fixed SNR value (30 dB), and test it for incoherent sources with different direction of arrival. It reveals that the spatial convolution beamformer have the same pattern as the DAS beamformer, but in comparison the pattern are shifted due to a narrower mainlobe compared to the DAS beamformer. The mainlobe is narrower due to the mentioned resolution advantage spatial convolution has, compared to DAS. This can lead to better performance for some directions of arrival for the DAS beamformer, which happens when the interfering source arrives from 37° degrees, shown in the figures (3.21). The Capon beamformer with subarray averaging have a significant higher performance and the results have a smoother variation, than the spatial convolution and the DAS beamformer. The performance of coherent sources (3.22), we observe that the spatial convolution method does not improve the SNR related metrics compared to Capon with subarray averaging. It does not either exceed the performance of the DAS beamformer, but stays at the same level. It will almost give the same result as in the incoherence case.

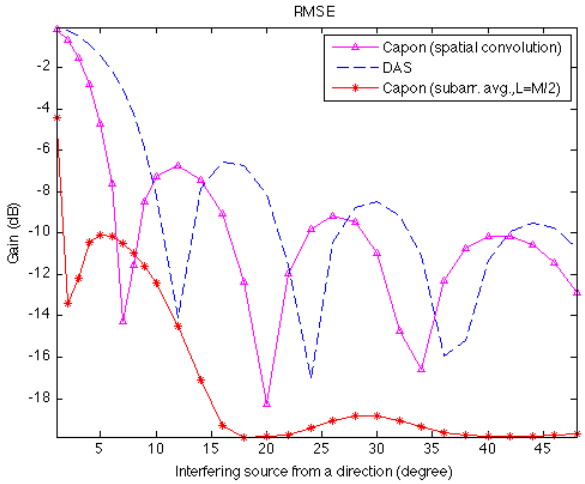
3.4.5 Number of Time Samples

Finite data samples are a common and known problem in construction of beamformers. When dealing with non- stationarties or varying characteristics of the statistics, will errors in the estimates occur. To be completely sure that the statistics always are correct we need an infinity number of data samples. But due to the problem of having infinity data samples, and because of the motivation of real-time processing we do not always have enough samples available. To figure out how dependent the spatial convolution method is to the number of time samples, we have studied the behavior of SNR metrics while changing N. We have made a wave field that consists of one desired source in angle 0°, and a coherent source that is varied from $-45°$ to 45°. To study the dependency of the number of time samples we have tested for $N = 1$ up to $N = 100$, which is ten times bigger than the spatial sampling size M. The SNR value is set to zero. We have chosen to compare the result with a Capon beamformer with subarray averaging, with a subarray length equal to $L = M/2$. The figure (3.23) shows that the Capon with subarray averaging has a more stable high SNR gain, and the spatial convolution varies more with its sidelobes.

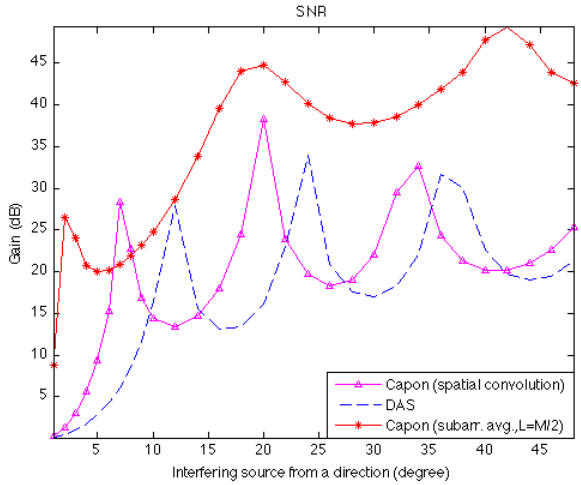
The figure (3.23) shows that the spatial convolution method needs more time samples to reach a stable performance of SNR, compared to the Capon beamformer. The blue area in the middle indicates the mainlobe, which is steered towards 0°.



(a) Array Gain

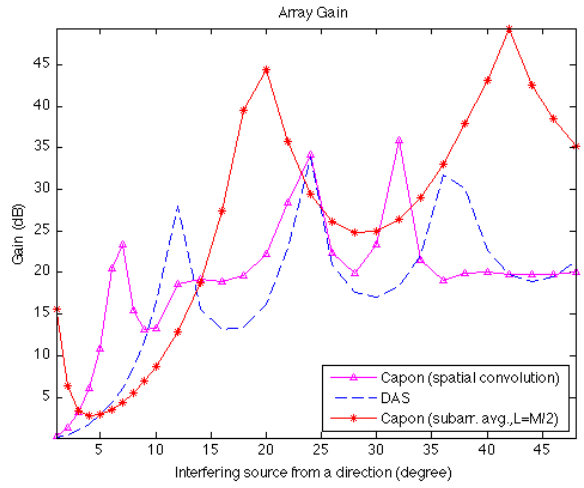


(b) RMSE

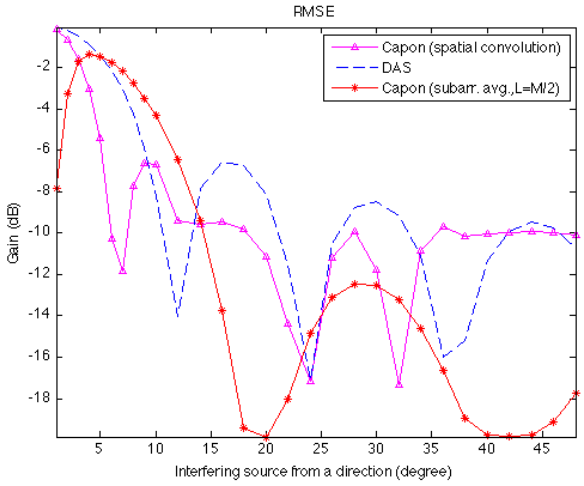


(c) SNR

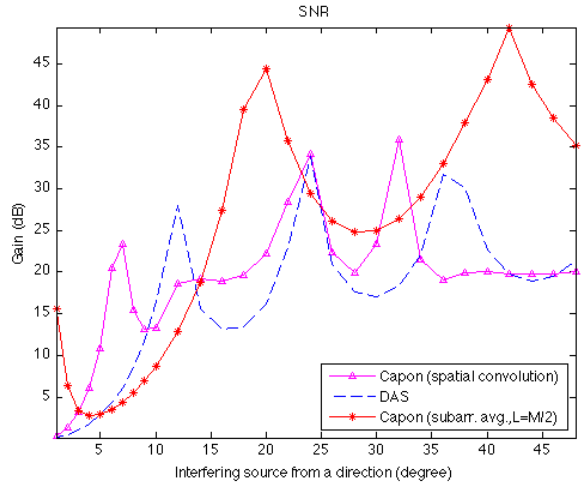
Figure 3.21: SNR related metrics for fixed SNR value (30 dB) and different direction of arrival of a incoherent interfering source.



(a) Array Gain

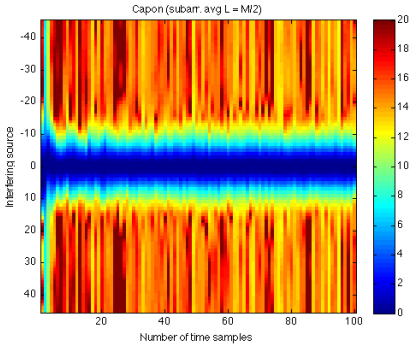


(b) RMSE

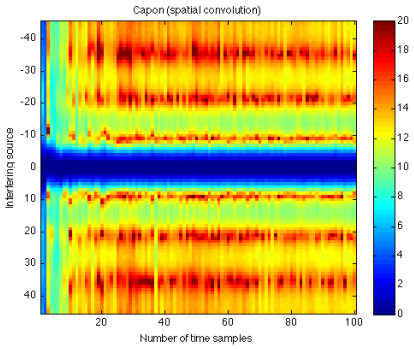


(c) SNR

Figure 3.22: SNR related metrics for fixed SNR value (30 dB) and different direction of arrival of a coherent interfering source.



(a) Capon (subarr. avg. L = M/2)



(b) Spatial convolution

Figure 3.23: Measuring the SNR (dB) for different time samples and different angle of interference

3.4.6 Attenuation Properties

We investigate the spatial convolution method attenuation properties by comparing it to Capon with subarray averaging. It shows that the spatial convolution method has a weaker performance than the Capon method when it comes to the ability to place zeros in the directions of interference. The Capon method is also better to attenuate the interference source in terms of gain.

Although, the spatial convolution method has a large number of degrees of freedom than the Capon method, since it uses a subarray length of $L = M$. The figure (3.24) illustrates a field with one coherent source and a desired signal. The coherent source is marked with green. Two sources in magenta is coherent with each other, but not with the desired signal. The black source is incoherent with all other sources. We have steered the beamformer towards $0°$, and we can read out form the plot that the distortionless response constraint in the steering vector is met for both methods. The illustration also shows the poor ability of the spatial convolution to place the zeros in direction of the interference. As mentioned, the spatial convolution method has a bias in the estimate direction of arrival, because the averaging makes the estimates inconsistent with the underlying signal model.

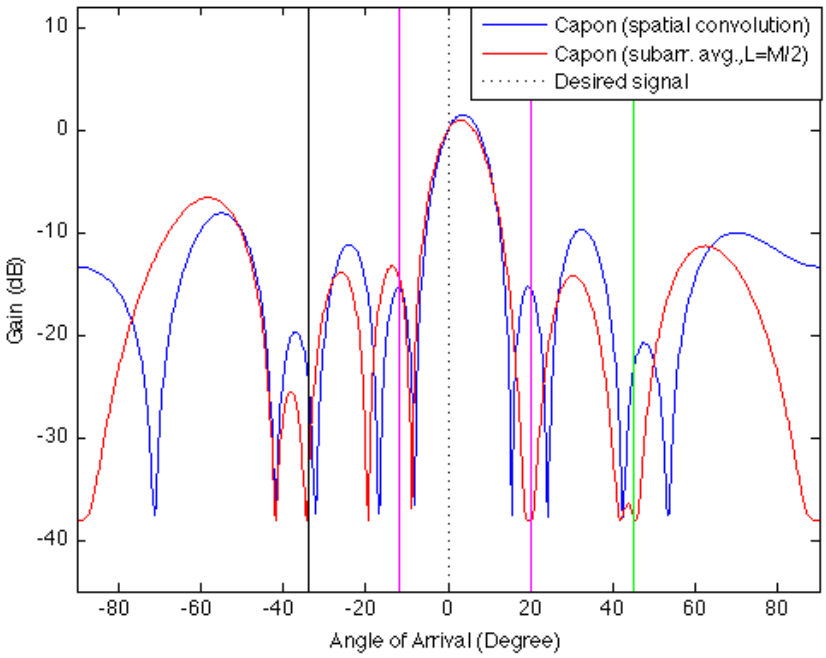


Figure 3.24: Beampattern of a random field using the spatial convolution method and Capon with subarray averaging.

3.4.7 Spatial Convolution Summary

The spatial convolution provides a good resolution better than DAS beamformer, but does not exceed the Capon beamformer using subarray averaging. The structure of the sidelobes of the spatial convolution method has similarities to the conventional DAS beamformer. It has a better resolution than the DAS, which causes a shift in the sidelobes pattern compared to DAS sidelobes. The spatial convolution has a weak performance when it comes to placing zeros in the direction of the interference, which is also reflected to the bias of the DOA estimates [11]. The numerical complexity of this method makes it highly relevant, it is very simple to implement, and the performance is better than the conventional techniques when it comes to resolution and suppressing coherent sources.

3.5 Comparison of the Toeplifing Methods

3.5.1 Resolution

When investigating the resolution of the Toeplitz constrained methods, we have made some assumptions that may affect the resolution. When measuring the resolution we have let the dip between the two sources grow by 3 dB for each measurement, this is one of the factors that may affect the result. The adaptive spatial averaging is as mentioned also constrained so the measured peaks has almost the same gain, otherwise

it would be difficult to measure the peaks. Another effect that had affected the resolution; is multiple sources in the wave field. We know that if several sources were present in the field they would have affected the results, but this applies to all of the tested beamformers. The resolution results

Beamformers	Resolution	
	Incoherent	Coherent
DAS	13.29°	16.90°
IAA-APES	3.83°	3.83°
Capon (Adaptive spatial avg., L = M/2)	5.18°	5.18°
Capon (Spatial convolution)	9.37°	10°
Capon (Subarray avg., L = M/2)	7.43°	10.14°
Capon	3.38°	17.8°

Table 3.4: Result of resolution for all tested beamformers

is based on a uniform linear array with ten sensor elements, $M = 10$. The table shows that the IAA-APES provides the overall best resolution performance. The full array Capon has a better resolution in the case of incoherence between the sources, but falls through in the case of coherence. IAA-APES and the adaptive spatial averaging are not affected of coherence and operate only with one resolution.

3.5.2 Direction of Arrival Error

For our Toeplified methods the performance of the estimate of the direction of arrival varies with how close the interference is to the desired source, and whether the signals are coherent or incoherent. As mention, the spatial convolution method operates with two different resolutions, in presence of coherent or incoherent sources. The two other methods, adaptive spatial averaging and IAA-APES, operates with only one resolution. All three methods have a bigger deviation in presence of coherent signals. The method that has the smallest deviation closest to the resolution threshold is the spatial convolution method, which has a deviation under one degree. But the spatial convolution is the one that varies most in the coherent case, the angle bias oscillate between 1° and 0° when the spacing between the desired and interfering sources increases. IAA-APES and the adaptive spatial averaging converge to a 0 in both cases of coherence. They converge to the minimum at approximately the same distant between the resolution threshold and the point where convergence threshold is reached. The adaptive spatial averaging has the largest deviation close to the resolution threshold in the case of coherent signals, it has an error up to 3° degrees.

3.5.3 Numerical Complexity

After analyzing our Toeplified methods we are able to rank the method by the numerical complexity. The method that have the lowest numerical complexity, and therefore perform best is the spatial convolution method,

which is also the simplest to implement. By just implementing the zero-padded the data vector $\vec{y}[n]$, and do a subarray averaging with subarray length equal the array length, $L = M$. The most time consuming task for the spatial convolution method is to invert the estimate of the covariance matrix, which require $O(L^2)$ operations.

The second method is the adaptive spatial averaging, which has two sets of weights to determine. The first weights v_k to calculate are for the estimate of the covariance matrix $\hat{\mathbf{R}}_{sa \cdot k}$, and the second set of weights, $\vec{w}$, are applied to the sensor data. Critical for this method is that we are not able to use the advantage of having an exact Toeplitz, so the two inversion have to be done in $O(K^3)$ and $O(L^3)$ operations, which are the most heavy operations in the process. The IAA-APES scores poorly on numerically complexity, as mentioned it is a iterative method and each iteration do the same operations as the other adaptive beamformers, including constructing the estimate of the covariance matrix, calculate the weights, and apply the resulting weights to the sensor data. As mentioned, the estimate that the IAA-APES method requires is about fifteen iterations to have satisfactory performance, which makes the algorithm worst. The result list is then presented;

1. Spatial convolution
2. Adaptiv spatial averaging
3. IAA-APES

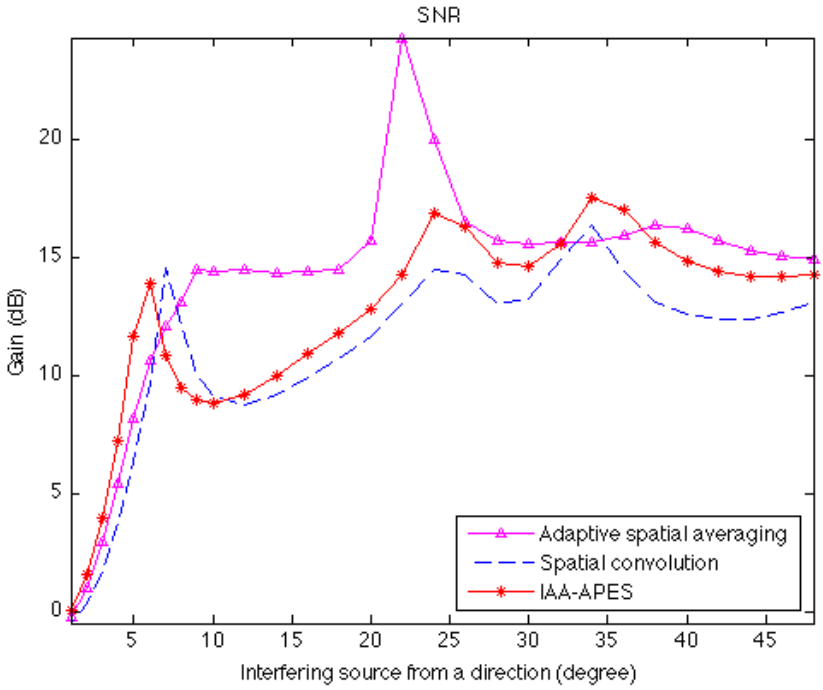
3.5.4 SNR related metrics

The adaptive spatial averaging is the Toeplified method that perform best on the SNR related metrics, it perform well for low SNR values, and outperform the other Toeplified beamformers when we set the a fixed SNR value for the simulated field to be 30 dB. The figure (3.25) shows the two cases, one low SNR value 0, and a SNR value of 30 dB.

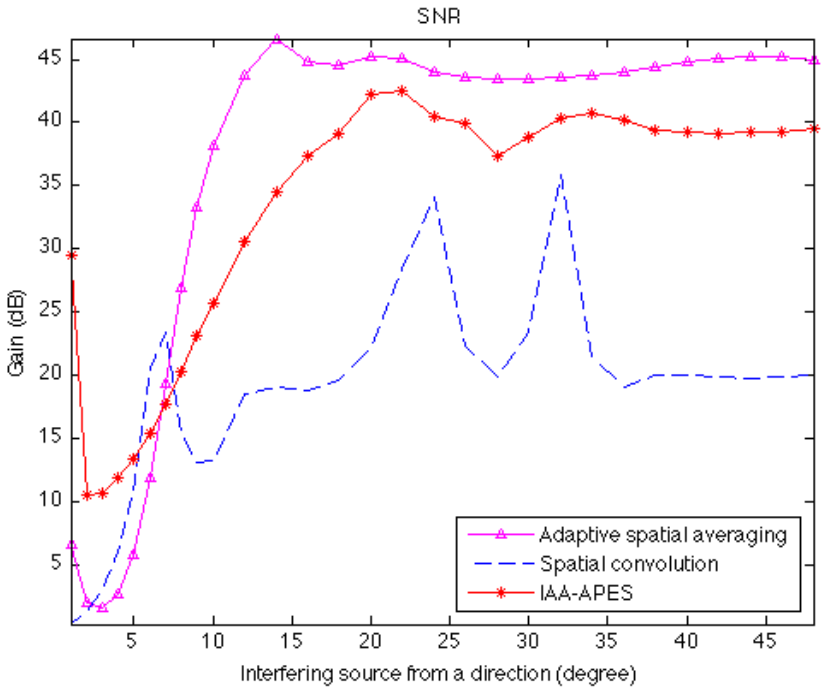
The IAA-APES perform well and have the same structure of the curve as the spatial convolution for SNR value 0, but has a higher gain. When the SNR is increased to 30 dB, the SNR performance of IAA-APES has differentiated from the spatial convolution. The spatial convolution performs similar to the DAS beamformer, but the sidelobes are shifted and give a better resolution.

3.5.5 Number of Time Samples

IAA-APES and the spatial convolution method need more time samples than the adaptive spatial averaging to reach its stable performance level. Whereas the adaptive spatial averaging needs approximately $N = 10$ time samples, the IAA-APES and the spatial convolution needs about $N = 20$. The explanation for this may be the size of the array or subarray. The desired stability is achieved when the number of time samples is twice as large as the array size. Which may can indicate that the number of time samples, N , is dependent on the size of the array, M, or subarray length L.



(a) SNR value 0



(b) SNR value 30

Figure 3.25: SNR value 0 and 30, the interference moves away from the desired signal in 0°

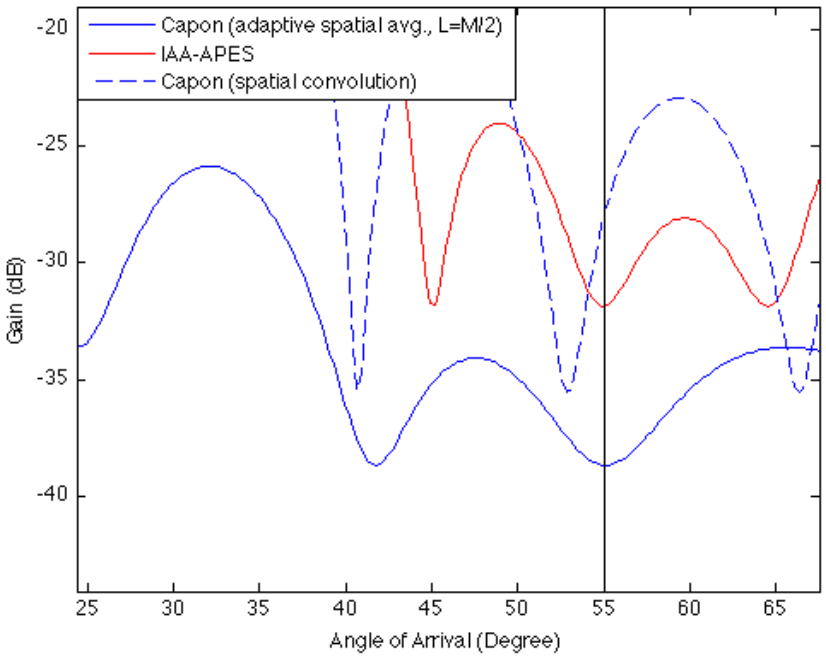


Figure 3.26: Comparison of the Toeplitz methods ability to place zeros in direction of the interference

3.5.6 Attenuation Properties

The first thing to notice is that both IAA-APES and spatial convolution has more degrees of freedom or more weights to suppress interfering sources than the adaptive spatial averaging method. Using the common subarray length $L = M/2$ the adaptive spatial averaging methods has $L - 1$ degrees of freedom, while the two other Toeplitz methods have $M - 1$ degrees of freedom. The adaptive spatial averaging has deeper suppression of the interference than IAA-APES and the spatial convolution method. The spatial convolution has the worst direction determination of interference, the zeros is not placed in the direction of the interfering source. The figure (3.26) is an example of the poor ability of the spatial convolution method to place zeros in the direction of the interference. It is the bias in the estimate of the covariance matrix $\hat{\mathbf{R}}_{sc}$, that makes the spatial convolution method is poor to determining the direction of the sources.

The sidelobes of the adaptive spatial averaging has a significant lower gain than IAA- APES and the spatial convolution.

3.5.7 The Frobenius Norm for the Toeplified Matrices

The Frobenius norm tells us how much the estimated structured Toeplitz matrix $\hat{\mathbf{R}}$ deviates from the real covariance matrix $\mathbf{R}$. The metric gives a benchmark for comparing the different Toeplified matrices, which then can be analyzed. Investigating the Forbenius norm metrics for given

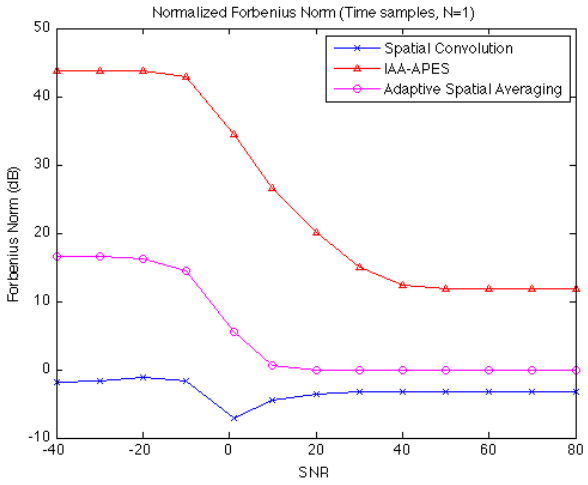
scenarios; changing the number of time samples N, and the SNR value of the wave fields. The tested case is steered towards 0°, and with a coherent interfering source in 40° degree. Since the curves are presented (3.27) in dB scale it means that if the two matrix are identical the dB value is minus infinity, $db(0) = -inf$. The figure (3.27) shows that the IAA-APES covariance matrix is not affected by the number of time samples, it perform approximately equal for every N that is investigated. The IAA-APES becomes closer to the true value of $\mathbf{R}$ when the SNR value of the wave fields increases, it is the beamformer that change the most. The spatial convolution is the method that is the most stable when it comes to the Forbenius norm, when $N = 1$ the estimated covariance matrix $\hat{\mathbf{R}}_{sc}$ moving towards the true value of $\mathbf{R}$ when the SNR increase. Otherwise, when $N = 10$ and $N = 20$ the estimate of the covariance matrix $\hat{\mathbf{R}}_{sc}$ deviates from the true covariance matrix $\mathbf{R}$. The behavior of the adaptive spatial averaging converges towards a dB value of 0 when the SNR increase. It is difficult to conclude anything from this results, but it shows that all methods converge towards the same result for every tested N. None of the beamformers is near the true covariance matrix.

3.5.8 Recent Work

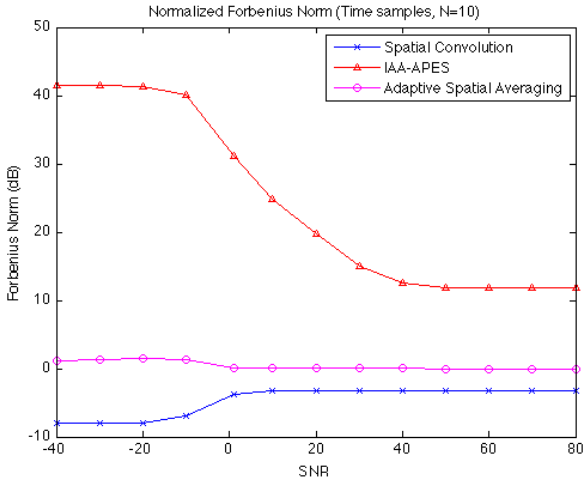
Asl and Mahloojifar published an article [3] within the same field of utilizing the Toeplitz structure of the estimated covariance matrix during the work of the thesis.They motivated the usage of the Toeplitz structure, due to the reduction of numerically complexity when inverting the covariance matrix. The structured matrix reduce the complexity of the inversion from $O(L^3)$ to $O(L^2)$, where L is the size of the subarray. The proposed method is to use standard subarray averaging to create an estimate of the covariance matrix ,$\hat{\mathbf{R}}$, which is then Toeplified by equalizing each subdiagonal with its averaged value. The results shows that the proposed method of Toeplified the estimates does not improve the performance, only reduces the numerically complexity of the inversion. It shows that the topic of this thesis is highly relevant, especially when adaptive weights shall be implemented in application such as in medically ultrasound imaging.

3.6 Applying Toeplitz Covariance Matrix Estimation to Ultrasound Imaging

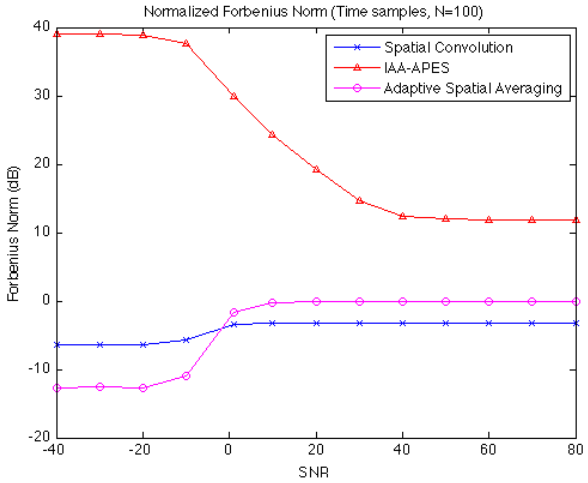
In order to test the Toeplified methods in the application of medically ultrasound imaging, we have made some simulations in the program Field II. Field II is a program for simulating ultrasound transducer fields and ultrasound imaging using linear acoustics [12, 13]. The simulation is done to verify the results, and show the performance of beamformers used in an application.



(a) N=1



(b) N=10



(c) N=100

Figure 3.27: Forbenius norm, the curves are normalized with the Forbenius norm of the true **R**. The beam is steer towards 0°, and a coherent inference in 40°.The illustration is in dB scale. 81

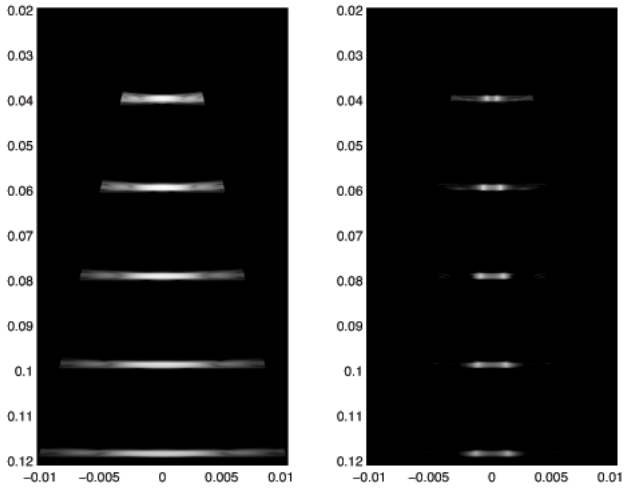


Figure 3.28: Images of point scatterers. **Left:** DAS. **Right:** Capon.

3.6.1 Point Scatterer

We did a comparison of the performance of IAA-APES, adaptive spatial averaging, and spatial convolution for separating point scatterers in ultrasound imaging system. We used an array of 64 elements, with a center frequency of 2.5MHz, and $\lambda/2$ element spacing. We placed 5 pairs of point scatterers at different depths, with an angular spacing of 0.7° between the scatterers.

The DAS image is shown on the left-hand side of the figure (3.28). The system is not able to resolve the point scatterers, and they appear as single scatterers. The Capon image is shown on the right-hand side of the figure (3.28), shows that all points are resolved.

A cross section of the point scatterers is shown in the figure (3.29), with the cross section of the point scatters the performance can easier be analyzed and compared. Here, we have also plotted the cross sections of IAA-APES, spatial convolution, and adaptive spatial averaging. It is clear that although Capon fully resolves the two points, but the method significantly underestimates the magnitude. In comparison, all the three suggested Toeplified methods yield better magnitude estimates, while still resolving the two sources. For the case of spatial convolution, the full 3 dB "dip" resolution is, strictly speaking, not achieved, but at least the points are separated.

3.6.2 "Generalized Diagonal Loading" From Spatial Convolution

Dynamic diagonal loading in ultrasound imaging is done as

$$\hat{\mathbf{R}} + \delta \text{trace}\{\hat{\mathbf{R}}\}/L\mathbf{I}, \tag{3.3}$$

The trace$\{\cdot\}$ function gives the sum of the main diagonal which is averaged by the number of inputs, L , on the diagonal. The identity matrix $\mathbf{I}$ ensures

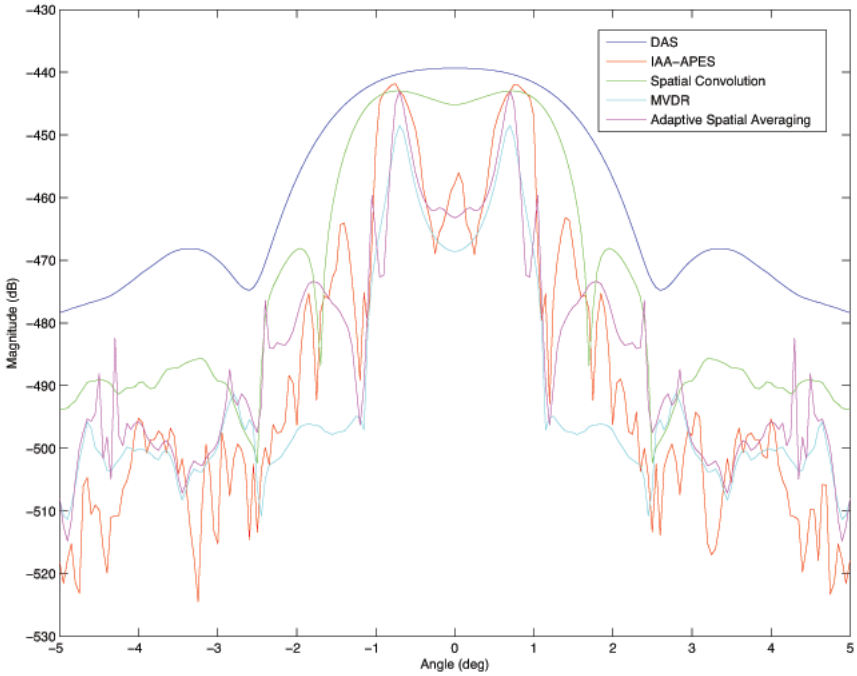


Figure 3.29: Cross section of point scatterers.

that the matrix always is invertible, and δ scales the identity matrix. As the scaling factor δ is increased, the solution of the Capon problem approaches DAS. The problem with this kind of robustification is that DAS is not an attractive solution performance-wise, except for its robustness. It is only natural to ask whether the dynamic diagonal loading principle can be generalized; can we add some scaled mean to each diagonal to increase the robustness? In other words; is there a Toeplitz matrix that can be added to $\hat{\mathbf{R}}$ to this end?

From [11], we can conclude that adding the mean of each subdiagonal will have little effect, because this matrix has been shown to yield similar Capon-weights as $\hat{\mathbf{R}}$ itself. However, we propose to make the Toeplitz structure by replacing each element in the subdiagonal with its subdiagonal summed and divided with the number of elements, M, i.e. the matrix used by spatial convolution, might work. The result is shown in figure (3.30), where we see that this form of generalized loading yields slightly better point separation than diagonal loading, while the attenuation on either side of scatterers is significantly worse. The sidelobe levels are about the same.

The contrast achievement is illustrated (3.31) with ultrasound image of a cyst, the DAS (3.31a) beamformer lets the speckle into the cyst (dark area), while the spatial convolution (3.31b) has reduced the leakage into the cyst. The Capon (3.31c) is the beamformer that perform best on speckle reduction inside the cyst, and it provide better edge detection of the cyst.

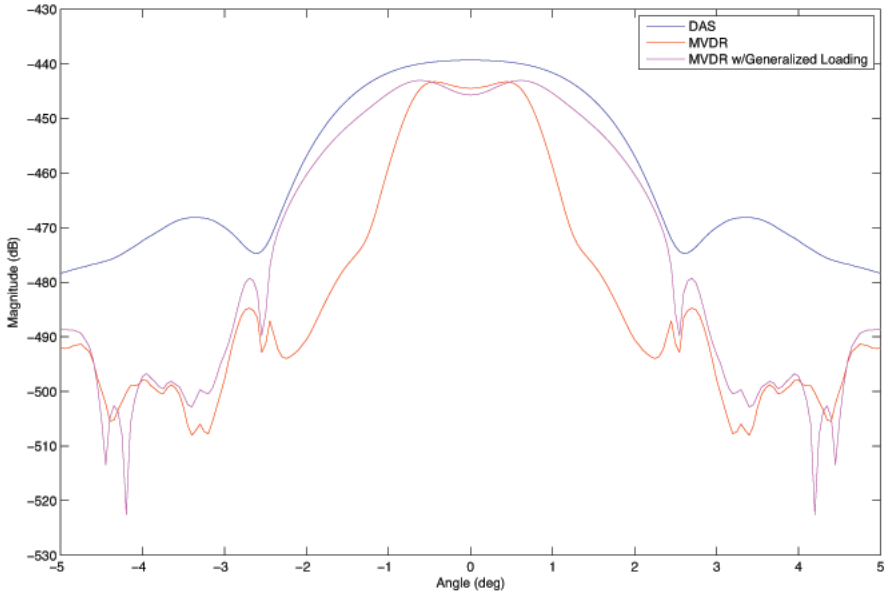
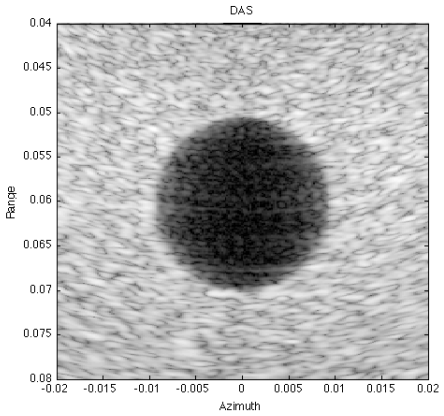
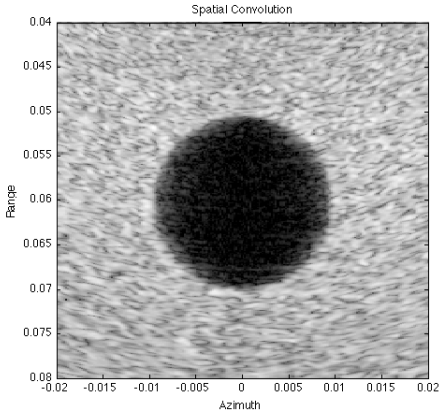


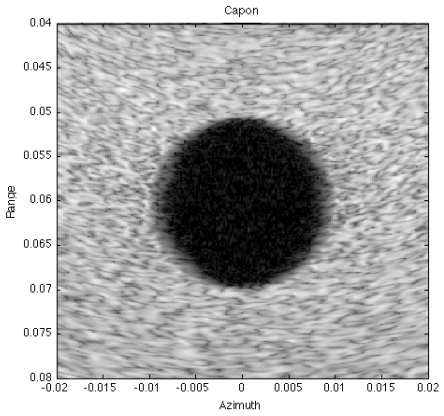
Figure 3.30: Cross section of point scatterers with generalized loading.



(a) DAS



(b) Spatial Convolution



(c) Capon

Figure 3.31: A simulated cyst in a medical ultrasound image, illustrating the contrast and edge detection achievement with the different beamformers; DAS, Spatial Convolution, and Capon.

Chapter 4

Discussion and Conclusion

During the investigation of the Toeplified beamformers has it been difficult to find a default scenario that does not favor one of the methods. More discussion of the number of interfering sources, its angle of arrival, the SNR value, number of time samples, coherent or incoherent signals, where to focus the beamformer, etc. is needed if searching for a single default scenario. However, while the testing in this thesis was performed using different parameters aiming to uncover discrepancies in the results, due to the randomness of wave fields no single default scenario should be able to cover all correctly.

The results of our Toeplified methods shows that they all contain good and less good aspects, which emphasizes that the choice is application oriented. The results show consistently that high performance comes at the expense of high numerical complexity. The highest resolution comes at the expense of the numerical complexity, which can make the investigated IAA-APES less attractive even though this method can be optimally implemented with parallel computing. The adaptive spatial averaging has a weakness because it is a semi-parametric beamformer that limits how many sources it is able to suppress. A semi-parametric beamformer need to know some priori knowledge about the field to work properly, as the adaptive spatial averaging have a constraint of the number of sources in the field. While the adaptive spatial averaging is the beamformer that preserves the underlying signal model best, both IAA-APES and spatial convolution have an inconsistent signal model in the estimate of covariance matrix. The effect is worse for the spatial convolution method where the inconsistent signal model leads to a bias in the DOA estimates, while the IAA-APES solves the bias problem with its iterations. The spatial convolution and the IAA-APES preserves its full array, while the adaptive spatial averaging reduces its size implying a reduction in the number of weights that can be used to suppress interference. The spatial convolution and IAA-APES are robust beamformers, while the adaptive spatial averaging is less robust due to the fact that the spatial convolution and IAA-APES always being Toeplitz, while the adaptive spatial averaging is only close to Toeplitz. The adaptive spatial averaging is not Toeplified because the method will preserve the underlying signal model. The

beamformer that has the simplest numerical complexity is the spatial convolution, and the robustness of spatial convolution can be compared to the DAS beamformer, which makes it attractive. Concluding of the preferred beamformer is difficult; what we can say is that all of them are dependent on the application.

As we discuss, the choice of beamformer is often a question between two different strengths. Is there a possibility of creating a beamformer combining two or several beamformers into one? The same principe as proposed in [26], the different is to choose adaptive weights instead of predetermined weights. The suggestion is to calculate two sets of weights, for example the standard APES and the spatial convolution weights. Both methods must eliminate the signal cancellation completely; otherwise it will cancel the desired signal.

$$\begin{aligned} \vec{w}_{apes} &= \min_{\vec{w}} \quad \vec{w}^H \hat{\mathbf{R}}_{apes} \vec{w} \\ \vec{w}_{sc} &= \min_{\vec{w}} \quad \vec{w}^H \hat{\mathbf{R}}_{sc} \vec{w} \\ P &= \min(|\vec{w}_{sa}^H \vec{y}[n]|^2, |\vec{w}_{sc}^H \vec{y}[n]|^2) \end{aligned}$$

The minimum of the two power estimates can be chosen because of the knowledge of distortionless response constraint, keeping unit gain in the steering direction, $\vec{w}^H \vec{d} = 1$. The idea of combining several beamformers' output is to improve the performance by choosing the best from the two methods. It is known that the spatial convolution method has a bias of the estimate of the direction of arrival, and the other beamformer could correct the bias. It is desirable that all beamformers shall complement each other, to gain the best result. The numerical complexity will increase, since it requires calculations for each adaptive weight. A solution to this problem is to choose one adaptive beamformer and one beamformer with predetermined weights, where the predetermined weights shall work as a safety net, so that the performance does not worsen. This is especially useful for methods such as spatial convolution, where the covariance matrix used in the minimization differs from the actual data. It is interesting to look further into which combination of beamformers will work best together?

There are several methods for making the covariance matrix Toeplitz, but we concentrated only on three due to the time constraints of the thesis. It could be interesting to compare other Toeplitz constrained to our results in order to improve understanding and get a broader picture of the Toeplitz constrained methods. The EM-algorithm or Expectation Maximization [18] is an example of a method that provides a Toeplitz constrained covariance matrix – the algorithm has a high numerical complexity. We know that redundancy averaging, where the elements on each subdiagonal are replaced with the mean value of its subdiagonal, for the complete $M \times M$ matrix, has similarities to the spatial convolution method. Therefore, how does the redundancy averaging perform on other metrics, such as SNR and resolution?

Another thing that can be investigated is how the IAA-APES and the

spatial convolution are related. After the first iteration of IAA-APES the result has similarities to spatial convolution, which is an interesting topic to look further into.

Appendix A

Wave fields generator

```
function [d,k,x,A,s,noise,amp_vec] = signalgenerator_dynamic(beamformerConfig)
%
% Generate 2 signals in noise according to page 73 of
% H. Krim, M. Viberg, "Two decades of array signal processing research
% — The parametric approach," IEEE Signal Processing Magazine, pp.67—94,
% July 1996.
%
% Note that definition of direction of arrival (DOA) is different from that
% of Krim and Viberg in that 0 degrees is normal incidence
%
% Written by:
% S. Holm, Department of Informatics, University of Oslo
% 31. Oct 1997 First version
% 21. Nov 1997 Corrected for noise in imaginary part 21. November 1997
% 12. Dec 1997 Added random phase noise to signals to better ensure incoherence
%  3. May 2011 James Trotter: updated setting of seed for random generator
%               for Matlab 7

% Modified Michael Pettersson: Can generate multiple signals


M = beamformerConfig.M;
N = beamformerConfig.N;
direction_vec = beamformerConfig.source_direction;
SNR_vec = beamformerConfig.snr_signal;
element_positions = (-(M-1)/2:(M-1)/2); % element postion Uniform linear array

gamma  = 45;                % degrees, phase shift between sources
d = 0.5;                % element spacing in array
k = 2*pi;               % normalized wavenumber
T = 0.5;                % sampling interval
omega1 = 2*pi;              % normalized frequency
omega2 = 2*pi;


% randn('seed',0);                      % replaced 2011 for Matlab 7:
RandStream.getDefaultStream.reset();    % make sure the same noise
                                        % sequence is generated each time
noise = randn(M,N) + 1i*randn(M,N);     % Gaussian noise, variance = 1.0
                                        % for real and imag part, i.e.
```

```
                                        % variance of noise = 2
amp_vec = zeros(size(direction_vec));
phi_vec = zeros(size(direction_vec));
a = zeros(M,length(direction_vec));
phase_vec = zeros(N,length(direction_vec));
all_signal = zeros(N,length(direction_vec));
clean_signal = zeros(N,length(direction_vec));

for ii = 1:length(direction_vec),

    amp_vec(ii) = sqrt(2)*10^(SNR_vec(ii)/20);
    phi_vec(ii) = -k*d*sind(direction_vec(ii));



    a(:,ii) = exp(1i*phi_vec(ii)).^element_positions';


    phase_vec(:,ii) = 1i*2*pi*rand(N,1);


    % Coherent or Non-Coherent
    if (beamformerConfig.coherent == 1) && (ii > 1)
           phase_vec(:,ii) = phase_vec(:,1);
    end
    all_signal(:,ii) = amp_vec(ii)*exp(1i*gamma*pi/180)*exp(phase_vec(:,ii))...
        .*exp(1i*omega1*T).^(0:N-1)';

    clean_signal(:,ii) = exp(1i*gamma*pi/180)*exp(phase_vec(:,ii))...
        .*exp(1i*omega1*T).^(0:N-1)';


    if ii == 1
        s = all_signal(:,ii);
        A = a(:,ii);

    else
        A = [A a(:,ii)];
        s = [s all_signal(:,ii)];

    end

end
s = s/max(amp_vec);
noise = noise/max(amp_vec);

s = s';

x = A*s + noise;
```

Bibliography

[1] Martin E. Anderson and Gregg E. Trahey. A seminar on k-space applied to medical ultrasound. http://dukemil.bme.duke.edu/Ultrasound/k-space/index.htm, 2006. [Online; accessed 12-May-2012].

[2] B.M. Asl and A. Mahloojifar. Contrast enhancement and robustness improvement of adaptive ultrasound imaging using forward-backward minimum variance beamforming. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 58(4):858 –867, april 2011.

[3] B.M. Asl and A. Mahloojifar. A low-complexity adaptive beamformer for ultrasound imaging using structured covariance matrix. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 59(4):660 –667, april 2012.

[4] Ann E. A. Blomberg. *Adaptive Beamforming for Active Sonar Imaging*. PhD thesis, University of Oslo, 2012.

[5] J.P. Burg, D.G. Luenberger, and D.L. Wenger. Estimation of structured covariance matrices. *Proceedings of the IEEE*, 70(9):963 – 974, sept. 1982.

[6] J. Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8):1408 – 1418, aug. 1969.

[7] D.R. Fuhrmann. Application of toeplitz covariance estimation to adaptive beamforming and detection. *Signal Processing, IEEE Transactions on*, 39(10):2194 –2198, oct 1991.

[8] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3rd ed. edition, 1996.

[9] D. A. Gray, B. D. O. Anderson, and P. K. Sim. Estimation of structured covariances with application to array beamforming. *Circuits, Systems, and Signal Processing*, 6:421–447, 1987. 10.1007/BF01600233.

[10] I.K. Holfort, F. Gran, and J.A. Jensen. Investigation of sound speed errors in adaptive beamforming. In *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, pages 1080 –1083, nov. 2008.

[11] K.C. Indukumar and V.U. Reddy. A note on redundancy averaging. *Signal Processing, IEEE Transactions on*, 40(2):466 –469, feb 1992.

[12] J. A. Jensen and N. B. Svendsen. Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, 39:262–267, 1992.

[13] Jorgen Arendt Jensen. Field: A program for simulating ultrasound systems. In *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1:351–353*, pages 351–353, 1996.

[14] D. H. Johnson and D. E. Dudgeon. *Array Signal Processing: Concepts and Techniques*. Pentice Hall, 1993.

[15] Richard J. Kozicka and Saleem A. Kassamb. A unified approach to coherent source decorrelation by autocorrelation matrix smoothing. *Signal Processing*, 45(1):115 – 130, 1995.

[16] Jian Li and P. Stoica. An adaptive filtering approach to spectral estimation and sar imaging. *Signal Processing, IEEE Transactions on*, 44(6):1469 –1484, jun 1996.

[17] Jian Li, P. Stoica, and Zhisong Wang. On robust capon beamforming and diagonal loading. *Signal Processing, IEEE Transactions on*, 51(7):1702 – 1715, july 2003.

[18] M.I. Miller and D.L. Snyder. The role of likelihood and entropy in incomplete-data problems: Applications to estimating point-process intensities and toeplitz constrained covariances. *Proceedings of the IEEE*, 75(7):892 – 907, july 1987.

[19] C.-I.C. Nilsen and I. Hafizovic. Beamspace adaptive beamforming for ultrasound imaging. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 56(10):2187 –2197, october 2009.

[20] Carl-Inge Colombo Nilsen. *Beamformers with Applications in Medical Ultrasound Imaging*. PhD thesis, University of Oslo, 2010.

[21] Tie-Jun Shan and T. Kailath. Adaptive beamforming for coherent signals and interference. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(3):527 – 536, jun 1985.

[22] P. Stoica, Hongbin Li, and Jian Li. A new derivation of the apes filter. *Signal Processing Letters, IEEE*, 6(8):205 –206, aug. 1999.

[23] Petre Stoica, Andreas Jakobsson, and Jian Li. Matched-filter bank interpretation of some spectral estimators.

[24] J. Synnevag, A. Austeng, and S. Holm. Minimum variance adaptive beamforming applied to medical ultrasound imaging. In *Ultrasonics Symposium, 2005 IEEE*, volume 2, pages 1199 – 1202, sept. 2005.

[25] J.-F. Synnevag, A. Austeng, and S. Holm. Adaptive beamforming applied to medical ultrasound imaging. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 54(8):1606 –1613, august 2007.

[26] J.-F. Synnevag, S. Holm, and A. Austeng. A low complexity data-dependent beamformer. In *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, pages 1084 –1087, nov. 2008.

[27] Johan-Fredrik Synnevag. *Adaptive Beamforming for Medical Ultrasound Imaging*. PhD thesis, University of Oslo, 2009.

[28] K. Takao and N. Kikuma. An adaptive array utilizing an adaptive spatial averaging technique for multipath environments. *Antennas and Propagation, IEEE Transactions on*, 35(12):1389 – 1396, dec 1987.

[29] P. Tourtier and L. Scharf. Maximum likelihood identification of correlation matrices for estimation of power spectra at arbitrary resolutions. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, volume 12, pages 2066 – 2069, apr 1987.

[30] Harry L. Van Trees. *Optimum Array Processing (Detection, Estimation, and Modulation Theory, Part IV)*. Wiley-Interscience, 1 edition, March 2002.

[31] F. Vignon and M.R. Burcher. Capon beamforming in medical ultrasound imaging with focused beams. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 55(3):619 –628, march 2008.

[32] F. Viola and W.F. Walker. Adaptive signal processing in medical ultrasound beamforming. In *Ultrasonics Symposium, 2005 IEEE*, volume 4, pages 1980 – 1983, sept. 2005.

[33] Zhisong Wang, Jian Li, and Renbiao Wu. Time-delay- and time-reversal-based robust capon beamformers for ultrasound imaging. *Medical Imaging, IEEE Transactions on*, 24(10):1308 –1322, oct. 2005.

[34] B. Widrow, K. M. Duvall, R. P. Gooch, and W. C. Newman. Signal cancellation phenomena in adaptive antennas: Causes and cures. *IEEE Trans. on Antennas and Propagation*, 30:469–478, 1982.

[35] D.M. Wikes and M.H. Hayes. Iterated toeplitz approximation of covariance matrices. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 1663 –1666 vol.3, apr 1988.

[36] T. Yardibi, Jian Li, P. Stoica, Ming Xue, and A.B. Baggeroer. Source localization and sensing: A nonparametric iterative adaptive approach based on weighted least squares. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(1):425 –443, jan. 2010.